

LOAD

OCTUBRE 1987
MSX

AÑO 2 Nº 18

₡5 REP. ARGENTINA

**BASIC PARA
MSX Y PC**

**MANEJO
DE PANTALLAS**

**COMO ACELERAR
UN PROGRAMA**

LENGUAJE C

CONCURSOS Y SORTEOS



EN INFORMATICA TALENT MSX HACE ESCUELA.

Leading

Y una prueba de ello, son algunos de los establecimientos que han incorporado computadoras Talent MSX como herramienta de apoyo pedagógico.

CAPITAL FEDERAL:

SIDERCA SAIC
ASOC. CRISTIANA DE JOVENES
COLEGIO JESUS MARIA
ESC. Nº 2 D.F. SARMIENTO
UNIVERSIDAD DEL SALVADOR
C.O.D.I.C.E.
CENEA
C.L.A.I.C.E.
INST. INMACULADA CONCEPCION
FUND. NTRA. SRA. DE LA MERCED
FUND. HNOS. A. Y E. ROCCA
INST. TECNICO DE BS. AS.
ESCUELA ARG. MODELO
COLEGIO ESTEBAN ECHEVERRIA
INST. JOSE MANUEL ESTRADA
ASOCIACION ISRAELITA ARGENTINA
INST. LA INMACULADA
ESC. Nº 24
CTRO. DE INF. PSICOPEDAGOGICA
NTRA. SRA. DE LA MISERICORDIA
ESC. REP. ORIENTAL DEL URUGUAY
ESC. Nº 10
ESC. MODELO D.F. SARMIENTO
INST. NTRA. SRA. DE LOS REMEDIOS
INST. PRIV. SAN CAYETANO
COLEGIO SAN GREGORIO
COL. MARIE MANOOGIAN
ESCUELA Nº 11
ESC. Nº 14 FRANCISCO BEIRO
INST. SAN VICENTE DE PAUL
ESC. Nº 11 POR LA NINEZ
INSTITUTO BAYARD
LAB. DE COMP. CLINICA Y EDUC.
ESC. Nº 5 URSULA DE LAPUENTE
COLEGIO ISLAS MALVINAS
COL. CHARLES DE FOULCAUD
C.O.E.S.O. LTDA.
NTRA. SRA. DEL SAGRADO CORAZON
ESCUELA ARGENTINA 2000
ESC. TEC. RAGGIO
BS. AS. ENGLISH HIGH SCHOOL
ESC. M. N. VIOLA
INST. SAN PIO X
ESCUELA Nº 5
INST. MARIA ANA MOGAS
INST. SUDAMERICANO MODELO
COLEGIO NTRA. SRA. DEL MILAGRO
CESCOM
COL. SGDO. CORAZON
COLEGIO SANTA BRIGIDA
INST. SAN PATRICIO
COLEGIO ANDERSEN
E.N.E.T. Nº 13
PROG. CLUT. EN SINDICATO
CIR SUBOF. DE LA POLICIA FED.
NEW MODEL SCHOOL

BUENOS AIRES:

INST. SAN GABRIEL - ADROGUE
ESC. ENS. MEDIA Nº 4 - ALGARROBO
ESCUELA Nº 28 - AVELLANEDA
INST. FRENCH - AVELLANEDA
E.N.E.T. Nº 1 V. PEREDA - AZUL
ESC. ENS. MEDIA Nº 4 - BAHIA BLANCA
COLEGIO DON BOSCO - BAHIA BLANCA
ESC. SUP. DE COMERCIO - BAHIA BLANCA
INST. SUP. JUAN XXIII - BAHIA BLANCA
CTRO. DE EDU. AVANZADA - B. BLANCA
SEARCH S.A. - BAHIA BLANCA
ESCUELA Nº 12 - BERAZATEGUI

JBS INFORMATICA - BERAZATEGUI
ESCUELA Nº 3 - BERISSO
COMPUSTEM - BERISSO
STA. MARIA DE LAS LOMAS - BOULOGNE
COLEGIO GOODSPELL - BOULOGNE
E.N.E.T. Nº 5 C. SARMIENTO -
CAPITAN SARMIENTO
ESC. Nº 9 NTRA. S. DEL CARMEN -
CARLOS CASARES
ESC. Nº 7 D.F. SARMIENTO - C. CASARES
E.N.E.T. Nº 1 - CARLOS CASARES
ESC. Nº 1 J. M. ESTRADA - C. CASARES
ESC. ENS. MEDIA Nº 1 - CHASCOMUS
CENTRO INF. ESC. Nº 5 - CHASCOMUS
COL. CORAZON DE MARIA - CHASCOMUS
COL. JUAN GALO DE LAVALLE -
CHASCOMUS
ESCUELA Nº 1 D.F. SARMIENTO -
CORONEL PRINGLES
COLEGIO CRISTO REY - DOCK SUD
ESCUELA Nº 28 - DON TORCUATO
ESC. ENS. MEDIA Nº 5 - DON TORCUATO
ESCUELA Nº 14 - ESCOBAR
COLEGIO JESUS MARIA - FCIO. VARELA
INST. LA SALLE - FLORIDA
ACADEMIA COMERCIAL BELGRANO -
GRAL. RODRIGUEZ
INST. GRAL. PACHECO - GRAL. PACHECO
INST. FADER - GRAL. PACHECO
INST. DE LOS SGDOS. CORAZONES -
HAEDO
COL. SHOLEM ALEJEM - HAEDO
E.N.E.T. Nº 5 - HURLINGHAM
CTRO. COMUNITARIO Nº 5 - ISLA MACIEL
ESC. EDUC. MEDIA Nº 7 - I. CASANOVA
ESC. CRISTIANA EVANGELICA - ITUZAINGO
INST. PRIV. A. LINCOLN - ITUZAINGO
E.N.E.T. Nº 1 - JOSE C. PAZ
INST. GRAL. J. DE SAN MARTIN -
JOSE C. PAZ
ESC. DE EDUC. MEDIA Nº 2 - JUNIN
INST. SUP. DE FORM. DOC. Nº 20 - JUNIN
COLEGIO MARIANISTA - JUNIN
ESC. DE CADETES GRAL. IRAMAIN -
LA PLATA
INST. ANTONIO PROBOLO - LA PLATA
ESC. ENSEÑANZA MEDIA Nº 1 - LA PLATA
FAC. CIENCIAS VETERINARIAS - LA PLATA
FAC. CS. NATURALES - LA PLATA
INST. INV. BIOQUIMICAS - LA PLATA
ESC. CONCIPIO VATICANO II - LA PLATA
COL. MARIA AUXILIADORA - LA PLATA
UNIV. NAC. DE LA PLATA - LA PLATA
INSTITUTO ATENEA - LANUS
INST. ECLESTON - LANUS
ESCUELA Nº 69 - LANUS
ESCUELA Nº 54 - LANUS
U.T.N. PACHECO - LOS POLVORINES
COL. SAN AGUSTIN - M. DEL PLATA
ESCUELA Nº 3 - MAR DEL PLATA
ESCUELA Nº 14 - MAR DEL PLATA
FUNDACION BOLSA DE COMERCIO -
MAR DEL PLATA
CTRO. NAC. ENS. INFORMATICA -
M. DEL PLATA
COL. STELLA MARIS - MAR DEL PLATA
COL. ALBERTO SCHWEITZER - M. DEL PLATA
ESCUELA Nº 67 - MAR DEL PLATA
ESCUELA Nº 62 - MAR DEL PLATA
ESCUELA Nº 27 - MAR DEL PLATA
ESCUELA Nº 31 - MAR DEL PLATA
C.E.F.A. - MAP DEL PLATA

JARDIN DE INFANTES MIS MANITOS -
MAR DEL PLATA
INST. SUP. DE EST. ADMINISTRATIVOS -
MAR DEL PLATA
INST. SAN VINTE. DE PAUL - M. DEL PLATA
JARDIN DE INFANTES Nº 2 - M. DEL PLATA
ESC. Nº 1 D.F. SARMIENTO - M. DEL PLATA
INST. DON ORIONE - MAR DEL PLATA
ESC. MAR DEL PLATA - M. DEL PLATA
INST. FAST COMPUTACION - M. DEL PLATA
ESC. ENS. MEDIA Nº 5 - MARTINEZ
STA. TERESA DEL N. JESUS - MARTINEZ
ESC. EDUC. MEDIA Nº 2 - MAYOR
BURATOVICH
ESCUELA MEDIA Nº 3 - MEDANOS
ESC. ENSEÑANZA MEDIA Nº 4 - MERLO
E.N.E.T. Nº 1 - MORENO
INST. SAINT THOMAS BECKET - MUNRO
ESC. Nº 14 H. IRIGOYEN - NECOCHEA
INST. ARG. DE IDIOMAS - NECOCHEA
ESCUELA Nº 42 - NECOCHEA
ESC. ARG. DANESA ALTA MIRA -
NECOCHEA
ESCUELA Nº 17 - OLAVARRIA
COL. CTRO. CULTURAL ITALIANO - OLIVOS
COL. LA ASUNCION DE LA VIRGEN -
OLIVOS
COLEGIO TARBUT - OLIVOS
INST. D.F. SARMIENTO - OTAMENDI
ESCUELA Nº 28 - PALOMAR
INST. JOSE MANUEL ESTRADA -
PELEGRINI
AC. SUP. DE COMER. HELLER - PERGAMINO
INST. CRISTO REY - QUILMES OESTE
COL. SANTO DOMINGO - RAMOS MEJIA
ESC. ARGENTINA DEL OESTE - R. MEJIA
INS. DE ENS. SUPERIOR - RAMOS MEJIA
INST. COMER. RANCAGUA - RANCAGUA
ESCUELA Nº 16 - REMEDIOS DE ESCALADA
ESCUELA Nº 30 - SALADILLO
COL. SAN FERNANDO - SAN FERNANDO
NTRA. SRA. DE LA UNIDAD - SAN ISIDRO
COL. CARDENAL SPINOLA - SAN ISIDRO
ESC. Nº 1 DR. COSME BECCAR - SAN ISIDRO
ESCUELA Nº 22 - SAN ISIDRO
INST. 20 DE JUNIO - SAN ISIDRO
CENTRO DE ESTUDIOS - SAN MARTIN
INST. NTRA. SRA. DE FATIMA - S. MANUEL
INST. SUP. DE FORM. DOCENTE Nº 42 -
SAN MIGUEL
ESCUELA JUANA MANZO - SAN MIGUEL
COL. PARROQUIAL STA. N. TRUJUI -
SAN MIGUEL
INST. SAN NICOLAS DE BARI - S. NICOLAS
E.N.E.T. Nº 1 - TANDIL
COLEGIO ECLESTON - TEMPERLEY
ESC. Nº 6 BME. MITRE - TIGRE
COLEGIO SAN RAMOS - TIGRE
ESC. AGROPECUARIA - TRES ARROYOS
E.N.E.T. Nº 1 - TRES ARROYOS
INST. JESUS ADOLESCENTE - 3 ARROYOS
E.N.E.T. Nº 1 - TRES ARROYOS
ESC. NAC. DE COM. M. BELGRANO -
TRENQUE LAUQUEN
ESCUELA Nº 8 - TRENQUE LAUQUEN
ESCUELA Nº 17 - TRENQUE LAUQUEN
ESCUELA Nº 2 - TRENQUE LAUQUEN
E.N.E.T. Nº 1 - TRENQUE LAUQUEN
ESC. Nº 5 C. VILLEGAS -
TRENQUE LAUQUEN
ESC. EDUC. MEDIA Nº 2 - VERONICA
INST. NUEVA ENSEÑANZA - VINTE. LOPEZ

INST. MIGUEL HAM - VICENTE LOPEZ
ESC. NAC. DE COM. M. BELGRANO -
VILLA BALLESTER
INST. NTRA. SRA. DE LOURDES - V. MADERO
E.N.E.T. Nº 1 J. NEWBERY - V. LUZURIAGA
INST. TEC. ALIMENTARIA - 9 DE JULIO
E.N.E.T. Nº 1 - 9 DE JULIO
ESC. MEDIA Nº 1 - 25 DE MAYO
E.N.E.T. Nº 1 - 25 DE MAYO

CHUBUT:

CTRO. PRIV. COMPUTACION EDUC. - TRELEW

CORDOBA:

COL. WILLIAM C. MORRIS - CORDOBA
COL. JESUS MARIA - LOS NARANJOS
INST. DE ENS. SUPERIOR - RIO CUARTO
CONVENTO DE SAN FSCO. - RIO CUARTO
COL. SAN BUENAVENTURA - RIO CUARTO
INST. DE 9º ENS. M. BELGRANO - SACANTA
S.E.S.F. COMPUTACION - SAN FRANCISCO
INST. JOSE PENIA - VILLA CABRERA

CORRIENTES

TALLER GAJUELO GAJUELO - CORRIENTES
ESC. N. S. M. MANTILLA - CORRIENTES
INST. LOOK AND LEARN - CORRIENTES

ENTRE RIOS

U.T.N. - C. DEL URUGUAY
CTRO. C. I. Y DE LA PRODUCCION -
C. DEL URUGUAY
E.N.E.T. Nº 2 - GUALEGUAY
FACULTAD DE BIOINGENIERIA - PARANA
ESC. INF. ENTRE RIOS - PARANA
E.N.E.T. Nº 1 - PARANA
U.T.N. PARANA - PARANA
ENET Nº 1 PASCUAL ECHAGÜE - PARANA

JUJUY

ESCUELA J. I. GORRITI - S. S. DE JUJUY
LA RIOJA
INST. ARG. DE E. SECUNDARIOS - LA RIOJA
ESCUELA GABRIELA MISTRAL - LA RIOJA

MENDOZA

ESC. NAC. DE COM. M. BELGRANO -
GODOY CRUZ
CENTRO INF. COMP. EDUCATIVA - MAIPU
INST. PADRE VASQUEZ - MAIPU
COL. VIR. DEL CARMEN DE CUYO - MAIPU
UNIVERSIDAD DE MENDOZA - MENDOZA
ESC. DE COMER. M. ZAPATA - MENDOZA
INSTITUTO PRAXIS - MENDOZA
INST. TECN. PRIV. T. EDISON - MENDOZA
INTERFACE CTRO. DE CAPACITACION -
MENDOZA
INST. COMERCIAL PIO X - TUNUYAN

MISSIONES:

S.M. DE PROM. DE LAS CIENCIAS - POSADAS
TALLER DE COMP. LAMPARITA - POSADAS

NEUQUEN:

ESCUELA Nº 11 - NEUQUEN
JAR. DE INFANTES PIMPINELA - NEUQUEN
ACT. G. ING. Y ARQUITECTURA - NEUQUEN
ESC. ENS. MEDIA Nº 32 - P. DEL AGUILA
CTRO. PROV. ENS. MEDIA Nº 3 - ZAPALA
CTRO. PRIV. DE INFORMATICA - ZAPALA

RIO NEGRO:

ESC. COMUN Nº 95 - GRAL. ROCA
ESC. Nº 168 FCO. RIVAL - GRAL. ROCA
ESC. COM. I. MALVINAS - GRAL. ROCA
COL. SECUNDARIO Nº 9 - GRAL. ROCA

E.N.E.T. Nº 1 - GRAL. ROCA
JAR. DE INFANTES PAYASIN - GRAL. ROCA
I.P.E.A. - GRAL. ROCA
ESC. Nº 71 S. MARTIN - S.C. DE BARILOCHE
INT. C.E.D.E.I. - S.C. DE BARILOCHE
COL. SEC. Nº 11 - VILLA REGINA
INST. NTRA. SRA. DEL ROSARIO - V. REGINA

SALTA:

INST. VACH - SALTA

SAN JUAN:

INSTITUTO BIOINGENIERIA - SAN JUAN

SAN LUIS:

INST. INFANTIL STA. CATALINA - SAN LUIS
INST. CAUSAY - SAN LUIS
INST. MASTERSOFT - SAN LUIS

SANTA CRUZ:

ESC. Nº 5 CAP. ONETO - PTO. DESEADO
COL. SEC. Nº 8 NACIONES UNIDAS -
PTO. DE STA. CRUZ

SANTA FE:

COLEGIO DE LOS ARROYOS - ROSARIO
INST. POLIT. SAN MARTIN - ROSARIO
SERVIRAMA - ROSARIO
COL. NAC. SAN LORENZO - ROSARIO
INST. NTRA. SRA. DE GUADALUPE
ROSARIO
MAGIC COMPUTACION - ROSARIO
COL. SALECIANO S. JOSE - ROSARIO
ESC. Nº 55 D.F. SARMIENTO - ROSARIO
E.N.E.T. Nº 638 - ROSARIO
E.N.E.T. Nº 623 - ROSARIO
COLEGIO CRISTO REY - ROSARIO
ESC. COMP. LICEO RO-NES - ROSARIO
INST. SAGRADO CORAZON - SAN JORGE
INSTITUTO CORDOBA - SANTA FE
ESC. EDUC. TECNICA Nº 2 - SANTA FE
ENET Nº 2 - SANTA FE
UNIV. NAC. DEL LITORAL - SANTA FE
ESC. EDUC. TECNICA Nº 2 - SANTA FE
INST. PRIV. I. DEL N. JESUS - SAN JUSTO
INST. SUP. Nº 21 - SANTA FE

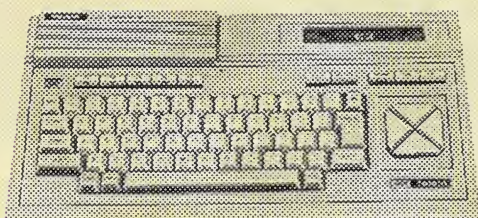
TUCUMAN:

INST. ING. SCHUAB Y TAPIA - TUCUMAN

Llene con sus datos el cupón al pie, envíelo por correo y recibirá en forma GRATUITA la revista INFORMATICA Y EDUCACION.

TELEMATICA S.A.

CHILE 1347 -
(1098) CAPITAL FEDERAL
TEL.: 37-0051/4



Talent
Tecnología y Talento
en el colegio

Nombre _____
Cargo _____
Establecimiento Educativo _____
Dirección _____
Provincia _____

Chile 1347 - (1098) Capital Federal

LOAD MSX

Director General

Ernesto del Castillo

Director Editorial

Cristian Pusso

Director Periodístico

Fernando Flores

Secretario de Redacción

Ariel Testori

Arte y Diagramación

Fernando Amengual y
Tamara Migelson

Departamento de Avisos

Oscar Devoto y
Nelso Capello

Servicios Fotográficos

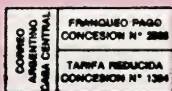
Image Bank, Oscar Burriel,
Victor Grubicy y
Eduardo Comesaña

Load Revista para usuarios de la norma MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5° Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Radiollamada: Tel.: 311-0056 y 312-6383, código 5941. Registro Nacional de la Propiedad Intelectual: E. T. M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de la Propiedad Intelectual. Todos los derechos reservados. ISSN 0326-8241

Precio de este ejemplar: \$ 5

Impresión: Calcotam, Fotocromo tapa: Columbia. Fotocomposición: Interamericana Gráfica.

Los ejemplares atrasados se venderán al precio del último número en circulación. Prohibida la reproducción total o parcial de los materiales publicados, por cualquier medio de reproducción gráfico, auditivo o mecánico, sin autorización expresa de los editores. Las menciones se realizan con fines informativos y técnicos, sin cargo alguno para las empresas que los comercializan y/o los representan. Al ser informativa su misión, la revista no se responsabiliza por cualquier problema que pueda plantear la fabricación, del funcionamiento y/o aplicación de los sistemas y los dispositivos descritos. La responsabilidad de los artículos firmados corresponde exclusivamente a sus autores. Distribuidor en Capital: Martino, Juan de Garay 358, P. B. Capital. Distribuidor interior: D G P: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800.

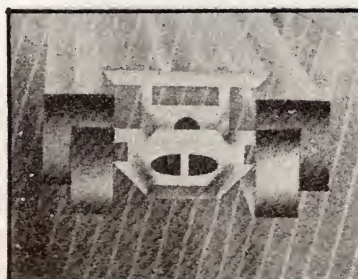


CONCURSOS



Se entregaron los premios del 2do. Concurso de Programas, y entrevistamos a Eduardo Blotta, ganador del 2do. premio. Además, anunciamos al ganador del concurso mensual "Notas, Trucos y Programas", y el cierre del concurso trimestral. (Pág.7)

COMO ACELERAR UN PROGRAMA BASIC (1ra. parte)



La velocidad de ejecución de los programas es importantísima para el usuario de microcomputadoras. En la carga de un juego o en los programas de comunicaciones, esta cualidad muchas veces es crítica. Para los que se enfrentan con este problema les brindamos las reglas que pueden solucionarlo. (Pág.8)

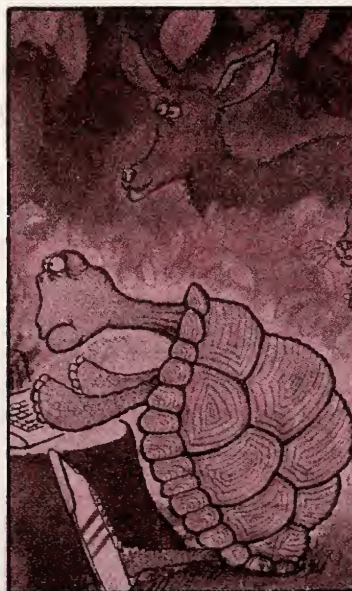
EL BASIC MSX Y EL DE PC



Nos referimos a las instrucciones iguales o de resulta-

do similar, que permiten usar un programa de MSX en una PC. (Pág.16)

RINCON DEL USUARIO



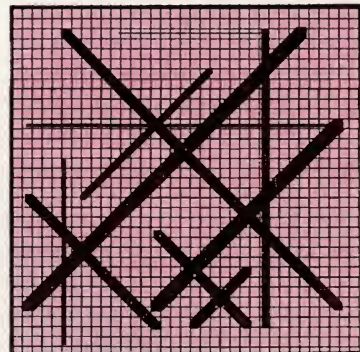
Explicamos el modo de encarar la enseñanza del Logo a través de una serie de preguntas-respuestas que clarifican y explican su alcance. (Pág.18)

LENGUAJE C PARA MSX: APLICACIONES



Presentamos el lenguaje C que está adquiriendo un gran auge en los círculos computacionales. Su velocidad y flexibilidad lo hacen ideal para la construcción de programas comerciales y sistema operativos. (Pág. 24)

MANEJO DE PANTALLAS EN MODO 0



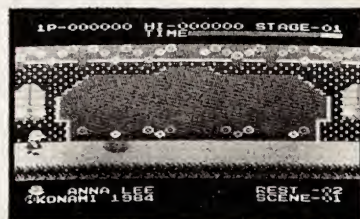
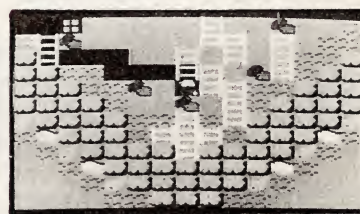
Para muchos, la norma MSX es la natural heredera de la TI-99/4A. Explicamos cómo a través de aquella se han rescatado y mejorado la mayoría de sus cualidades como por ejemplo las facilidades para realizar gráficos desde BASIC en modo 2 de alta resolución. (Pág.28)

PROGRAMAS



Do-Re-Mi (Pág.12) -Scrash (Pág.20)

SECCIONES FIJAS



Noticias MSX (Pág.4) - Soft al día (Pág.31) - Buzón (Pág.34)

"LO UNICO QUE IMPORTAMOS SON LOS CHIPS"

Con el lanzamiento de la MSX2 se puso de manifiesto la autonomía de Telemática respecto a la dependencia de proveedores del exterior. Carlos Manzanedo, director de la empresa, se refiere al tema.

puedan ser proveedores nuestros."

-¿Y con respecto al diseño de la plaqueta?

-En el caso de la MSX2, es un producto totalmente de-

gún tipo de dependencia de proveedores del exterior.

Advirtió que hoy pueden seleccionar cuándo les conviene comprar en Japón, cuándo en Estados Unidos, y cuándo en cualquier otro país que ofrezca determinados componentes a un mejor precio.

O sea que realmente uno pierde los complejos en lo que respecta al proceso de desarrollo industrial -agregó-. En la fabricación de computadoras esto es posible. Tan posible que estamos a precios competitivos y nos permite entrar nuestras computadoras en mercados prácticamente libres como es el de Chile, donde los recargos para importación de computadoras no son más del 20%. Competimos con productos que vienen de Estados Unidos, Europa y Oriente, y no estamos teniendo problemas de precios.

yar un fuerte desarrollo de la industria informática local que después permitiera encarar el mercado exportador. Lamentablemente, cuando salió esta resolución nosotros ya estábamos con nuestro propio proyecto muy avanzado en la provincia de San Luis por lo que no podíamos desarrollar la planta y reeditarla en alguna de las zonas promocionadas. Pese a no tener los beneficios que otorgaba la 652 nosotros seguimos adelante con nuestras propias convicciones, con vocación industrial. Si bien en estos momentos se están inaugurando esos dos polos informáticos, con lo cual recién ahora tendríamos que empezar a pensar cómo hacer el proceso de integración, nosotros ya hemos hecho lo que pedía la 652: una integración máxima en dos años, ¡pero nosotros la logramos en uno! Cuando pedían el lanzamiento de un producto de tecnología propia que no significara ningún pago por transferencia de tecnología al exterior en un plazo de dos años, nosotros lo estamos haciendo en un año y medio. A su vez se exigía un plan exportador a partir de los dos años y nosotros también en menos de dos años ya empezamos a exportar. Y esto cumpliendo en un todo con lo que pide la 652 pero con una ventaja: cuando los que adhirieron a la 652 recién están empezando, nosotros



Carlos Manzanedo

como la fuente, el transformador, la carcasa con la aparición de la MSX2 se entrega con matricería totalmente local. Hemos decidido, por ahora, seguir importando los teclados porque hasta el momento no hay ninguna empresa en el país que produzca teclados, ni a un precio razonable ni a una calidad que sea aceptable para el gran uso que hace de ellos el público al cual va dirigido. Nosotros queremos ser muy respetuosos en cuanto a la calidad del producto que ofrecemos. Con las industrias que se van a radicar en los polos informáticos posiblemente se desarrolle una industria buena de teclados y ellos

sarrollado por Telemática desde cero. Se arrancó desde una hoja en blanco, se han seleccionado cuáles son los componentes que más tecnología han incorporado en el mercado. Esto se puede ver en la plaqueta en donde hay un componente que tiene 100 patas, que es la última tecnología disponible en cuanto a componentes de silicio. Hemos incorporado y hemos ido seleccionando de distintos proveedores del mercado, los componentes más convenientes para elaborar un producto con la máxima prestación y con el costo más bajo. Eso nos permite hoy poder producir una computadora sin tener nin-



MSX2 Turbo

-Generando divisas... era lo que se quería con el Plan Informático Argentino.

-Exacto. O sea, el objetivo del Plan Informático, que estaba apoyado en la Resolución 44 y después se cambió por la 652, era apo-

ya nemos cubierto todas las etapas.

-Con respecto al servicio, a la parte técnica, ¿qué índice de fallas han detectado?

-Bueno, quizá suene como que son argumentos nuestros, de fabricantes, pero

basta con consultarlo en el mercado. Se puede ver que nuestros productos realmente no tienen más de un 2% de fallas. Estamos sorprendidos frente a nuestras propias estimaciones. Recomendaciones originales de los que han desarrollado la **MSX1** estimaban un promedio de fallas del orden del 2 al 3% y nosotros creíamos que al pasar a la producción local podíamos estar entre el 3 y el 5%. A un año de tener la producción local, -nosotros entendemos por producir localmente no sólo poner cuatro tornillos, sino armar la plaqueta, proveer una plaqueta industria nacional- estamos manteniendo el mismo nivel de falla que cuando importábamos la plaqueta armada, en los primeros meses del año pasado.

-Y ustedes tienen servicio técnico propio.

-Nosotros tenemos una red de más de 40 centros de servicio técnico en todo el país. Son los que simplemente atienden fallas ocasionales, que a veces se producen por un maltrato dado a la máquina o a fallas mínimas, como lo demuestra ese 2% de fallas.

-Y bastante rápido ¿verdad?

-Sí, eso es parte de la consigna que dice que cuando no se puede arreglar algo en el momento, se busca hacer el reemplazo inmediato de la parte que puede estar fallada y después en laboratorio se busca con más detenimiento la falla.

-Esto ¿en el service central o en todos?

-Básicamente en casi todos; podemos decir que el 80, 90% de los servicios técnicos están trabajando de la misma manera.

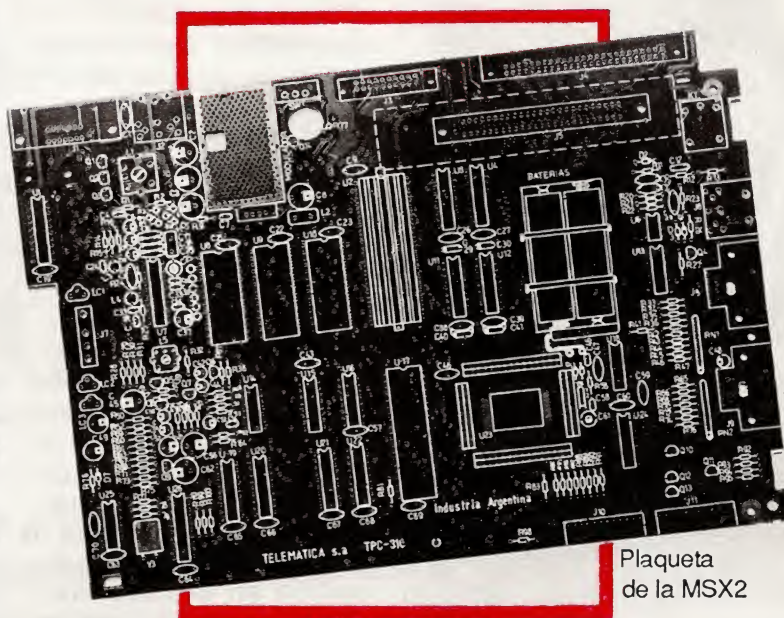
-Que dicen los orientales, obviamente no los de Uruguay sino los de Oriente.

-Bueno, realmente los ha sorprendido mucho. Nosotros estamos en trato permanente con los crea-

los que colocan su marca en otros países. Ellos están trabajando con gran cantidad de periféricos pero ninguno desarrollado bajo tecnología propia. Causa asombro que nosotros estemos ya sacando una **MSX2** a la que nos hemos dado el gusto de llamar **MSX2 Turbo** porque es la **MSX2** más veloz que existe en el mundo. En cuanto a la selección de componentes, por haberlos producido prácticamente a 9 meses de

cartuchos de 80 columnas que hay en el mercado sino que hemos entrado en un proceso de desarrollo y de miniaturización, porque ha habido algunos otros proveedores que han hecho cartuchos de 80 columnas pero con un tamaño que es el doble del nuestro y a un costo superior. Además desarrollamos la **Mini-Lan** que es la red local para uso educacional. Si bien existen otras redes educacionales en el mercado, están basadas en usar un equipo PC de 16 bits que vale muchísimo más que nuestro equipo central que es otra **MSX** cualquiera de las que está integrada a la red. Y sobre todo es la red local de uso educacional más económica que existe en el mundo.

Los ha sorprendido muchísimo el desarrollo propio que hemos hecho de un modem de características únicas en el mercado mundial en cuanto a poder incorporarles soft inteligente como puede ser todo el soft de comunicación incluido dentro del modem, el **Multiplan** (que es el **MSX-Plan**) y un procesador de palabras desarrollado por ASCII de Japón traducido totalmente por nosotros en el país. También los sorprende que hayamos desarrollado el teclado numérico. Para eso contamos con nuestro Departamento de Ingeniería, con más de 15 ingenieros entre los que se dedican al área de software y a la de hardware.



Plaqueta de la MSX2

dores de la norma **MSX**, que es la firma ASCII de Japón, y ellos están sorprendidos por el desarrollo tecnológico que se hace en Argentina. Tan sorprendidos que hay empresas del nivel de Philips, por ejemplo, u otras marcas, que ni siquiera tienen tecnología propia sino que compran productos a

diferencia con respecto al resto del mercado, nos permitió incorporar tecnología mucho más moderna y hemos logrado un mejor rendimiento y un más bajo costo. También los ha sorprendido cómo hemos hecho el desarrollo para la **MSX1** del cartucho de 80 columnas. No el desarrollo de otros

EL PAPELEO PARA SU MSX, LAS VENTAS PARA USTED.

Ahora, organice su empresa en forma rápida, simple y económica con nuestros sistemas, y usted sólo dedíquese a vender

MSX STOCK

LISTADOS
Artículos ✓ Precios
Stock valorizado ✓ Movimientos
Estadísticas de venta
EMISION DE
ROTULOS AUTOADHESIVOS

APLICACIONES

ORDENAMIENTO
POR
Categoría ✓ Número
Descripción ✓ Marca
Catálogo ✓ Proveedor

IMPRESION
Factura
Remito
Nota de débito
Nota de crédito

DISEÑADO ESPECIALMENTE PARA

Control de existencias (500 artículos)
Actualización de precios por rango y categoría.
Detalles de movimiento por artículo (3500 movimientos)
Manejo de IVA.

MSX CTAS CTES

LISTADOS
Clientes ✓ Saldos
Comprobantes ✓ I.V.A.
Convenio multilateral
Vencimientos ✓ Deudores
EMISION DE MAILING

APLICACIONES

ORDENAMIENTO
POR
Nombre
Código postal
Vendedor
Categoría
Número

IMPRESION
Factura
Remito
Nota de débito
Nota de crédito

DISEÑADO ESPECIALMENTE PARA

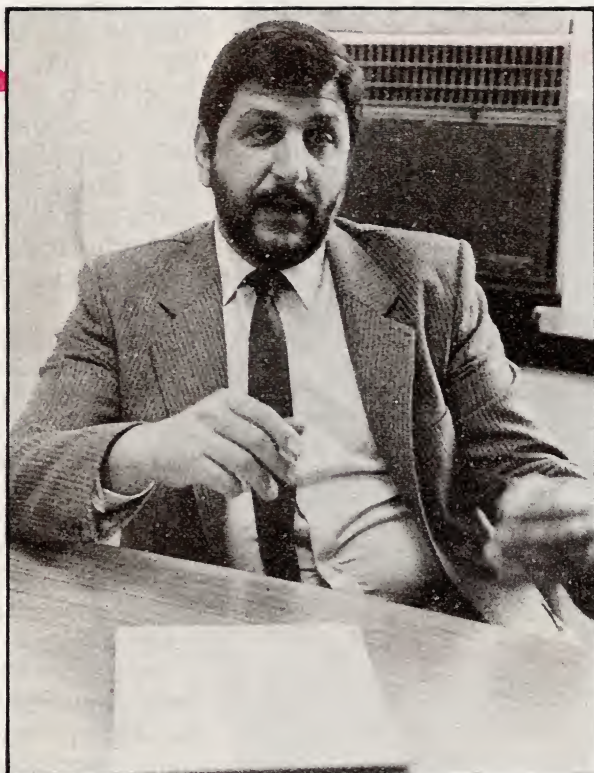
Carta de clientes.
(500 cuentas)
Detalle de movimiento por clientes (3500 movimientos)
Manejo de I.V.A.

SIMPLES Y RAPIDOS
Clave secreta de acceso
Constante orientación en pantalla
Validación de datos.
Confirmación de operaciones
Rápida corrección de errores
Manual de operación incluido

UNA EXCELENTE RESPUESTA

En la expansión del hardware producida en el mercado local, de la que se habla en esta misma página, el software educativo no ha quedado afuera. Para conocer características del incremento de su comercialización entrevistamos a **Antonio Giliberto**, director de Systemac.

Antonio Giliberto



¿Cómo fue la penetración de este software educativo?

-Bueno, la respuesta ha sido excelente. Si teníamos alguna duda al principio, realmente el usuario se está empezando a cansar del "mata marcianos", como yo los llamo. Quiere un software un poco más útil que aquel en el que simplemente toca un joystick, para disparar misiles y jugar. Realmente que sea más útil y que vaya formando al chico, para que empiece a entender desde una corta edad la importancia de la consola como una herramienta. Obviamente ningún software va a desarrollar la inteligencia del chico pero sí le va a empezar a introducir la filosofía de la computación que no se hace matando marcianos.

-¿Cuántos títulos tienen ya?

-Tenemos alrededor de 60 educativos ya, casi todos de procedencia europea (excepto dos que son desarrollo local), de firmas como **Dimensión New**, **Fisher Price**, **Idealogic** y **Spinnaker**.

-¿Y los que son de desarrollo local, cuáles

rie que son dos programas y se tiene muy en claro lo que es la lógica simbólica.

-¿Cuáles son los proyectos que tienen?

-Lanzaremos software educativo para la MSX2 fundamentalmente. Va a salir una serie de unos 25 títulos realmente educativos.

-¿Qué nos puede anticipar?

-Por ejemplo, había sobre física, Gases y Espejos Claros, que también está para MSX1, con otra resolución de pantalla, con otro tipo de software. Si bien la concepción, es decir la materia es la misma, es un poco más profundo porque la herramienta lo permite.

-¿También son del mismo origen?

-Sí.

-¿Para MSX1 van a seguir saliendo títulos?

-Sí. Nosotros estamos sacando al mercado un promedio de 8 títulos por mes.

-¿Qué perspectivas le ven al software educativo en el mercado?

-Yo le veo muy buena perspectiva. Día a día se venden más educativos y no pasa lo mismo con la venta de juegos, que está estática.

-En cuanto al desarrollo local ¿pensaron encarar algo nuevo?

-Sí, hay un par de casas de software que están traba-



jando. Estamos esperando que lo terminen para poder revisarlo y ver si cumplen con las expectativas que nosotros tenemos.

-Y en relación con el software de aplicación ¿en que están trabajando?

-Estamos trabajando con un paquete de gestión ventas, contabilidad, sueldos, video-clubes, administración de propiedades; se está terminando un paquete para inmobiliarias. Todos desarrollos locales. Evidentemente el soft de gestión tiene que ser de desarrollo local.

son ?

-Son uno de geografía y otro de matemáticas, muy curricular, que no han tenido muy buena acogida por ser muy estáticos. En cambio los otros son juegos educativos.

-Pero con respecto a los que son juegos educativos ¿ahí sí han tenido respuesta?

-Del docente y del alumno. Una excelente respuesta. Se han cuadruplicado las ventas en cinco meses.

-¿Responde a la curricula escolar ese software?

-No en forma directa, no es curricular, pero sí responde a cada materia específica. Por ejemplo, hay una serie que es **Logicolor** que es lógica metodológica y no sigue ninguna curricula de acá, pero se termina esa se-

PARA VIDEO CLUBES

Un software diseñado y desarrollado para manejar toda gestión de un video club tipo es distribuido por **Organización Maro**. Ideal para cualquier computadora de norma MSX, el sistema está pensado para crecer junto con la empresa. Así, la versión estándar, que maneja **1500 títulos y 500 clientes**, con el agregado de un disquete podrá ampliarse a **4000 títulos y 1300 clientes**. Además de las posibilidades de realizar estadísticas, mailing y seguimineto de películas y clientes, que cubre la actividad del club en los últimos seis meses, el sistema o-

frece una serie de características para el mejor funcionamiento del negocio, como por ejemplo: altas, bajas, consultas y modificaciones de películas, listados de títulos por géneros, control y listado de reservas, de deudores de películas y facturas, etcétera. (ver página 33).

Por otra parte, en lo que se refiere a sistemas generales, **Organización Maro** ofrece Contabilidad General, Stock y Cuentas Corrientes; y en el rubro de sistemas específicos presenta Administración Odontológica y Consultorios Médicos.

EL ROSTRO DEL SINTETIZADOR DE VOZ

Eduardo es casi un ingeniero electrónico (le faltan rendir solo cuatro finales), que trabaja en la Facultad de Ingeniería de Mar del Plata haciendo desarrollos de hardware y software.

Participó en febrero de un curso de Microelectrónica dictado por EBAI (Escuela Brasileiro Argentina de Informática), en Tandil y en e-



Manuel R. Rojas

nero próximo asistirá a Curitiba a tres seminarios sobre Inteligencia artificial, sistemas expertos y control digital.

El programa se originó cuando comenzó a buscar un tema para el proyecto final de su carrera. Después de desechar algunos, finalmente se decidió por el que lo estuvo tentando por varios años: el sintetizador de voz.

Como etapa previa al desarrollo del hard, encaró el problema intentando realizar el sintetizador totalmente en software, de esa manera alcanzó un resultado muy interesante.

Un factor que tuvo en cuenta fue almacenarlo en una zona de memoria, que lo

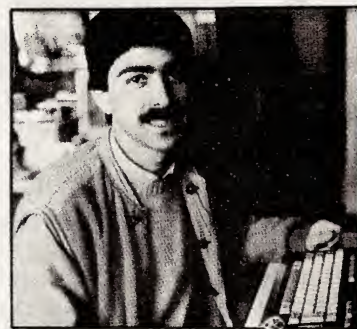
LOS GANADORES

Telemática S.A. entregó los premios a los ganadores del 2do. Concurso de Programas. Manuel Rojas, quien resultó vencedor con su programa Do-Re-Mi, ya fue presentado en el número anterior. Esta vez nos visitó Eduardo Luis Blotta que con su Sintetizador de Voz (publicado en el número 14, de Junio) obtuvo el segundo premio del concurso.

vuelva "transparente" al BASIC, es decir, que se inserta en la computadora con una instrucción más.

De visita en nuestra redacción, Eduardo se mostró entusiasmado por el lanzamiento de la MSX2, y anticipó su "salto" a esta máquina en un futuro cercano.

Sobre la utilidad del programa, nos comentó que, además, el proyecto, lo pensó con fines educativos, para acercar la informática a los no videntes, de manera que



Eduardo L. Blotta

aquellos que no puedan ver la pantalla puedan en cambio escucharla.

GANADOR DEL CONCURSO MENSUAL

Fue Miguel Luis Kopeck con su trabajo Dibujando en la MSX. La nota es una introducción al uso de instrucciones gráficas. Presenta una serie de programas que permiten dibujar con línea llena, punteada y

borrar. Además, brinda la posibilidad de variar las funciones cambiando algunos números o instrucciones dentro de los programas. El trabajo se publicará en el próximo número.

CIERRE DEL CONCURSO LOAD MSX

Recordamos a los que quieren participar del Concurso de notas, Trucos y Programas que el 15 de noviembre próximo finaliza el certamen. Por lo tanto, en el número de noviembre se conocerá el segundo ganador del concurso mensual, y el 15 de diciembre el ganador del premio trimestral.

Para quienes todavía no han participado, los requisitos son que los programas o trucos deberán servir para cualquier computadora de la norma MSX.

El premio mensual es un programa provisto por Telemática, a elección entre Idea Base, Idea Text o Basic Tutor.

El premio trimestral es un software de aplicación provisto por Telemática con un manual pudiendo optar entre el MSX-LOGO, MSX-PLAN o MSX-WRITE.

Los trabajos se deberán enviar a nombre de CONCURSO MENSUAL LOAD MSX, Paraná 720, 5º Piso, (1017) Cap. Fed.

SI! QUIERO LO MEJOR

¿Quien dispone de todo el soft para MSX ?

MAIL SOLUTIONS

1ª organización de venta por correspondencia de software (utilitarios y juegos) para MSX

Complete, recorte y envíe su ficha a:

Casilla de Correo nº 40
C.P. 1826 Buenos Aires

A vuelta de correo recibirá **GRATIS** nuestro catálogo y un obsequio sorpresa.

Nombre y apellido :

Edad : Actividad:

Dirección :

C.P. : Localidad :

Provincia :

Consola (marca) :

Disketera :

Datasette :

Impresora :

COMO ACELERAR UN PROGRAMA BASIC - PARTE I

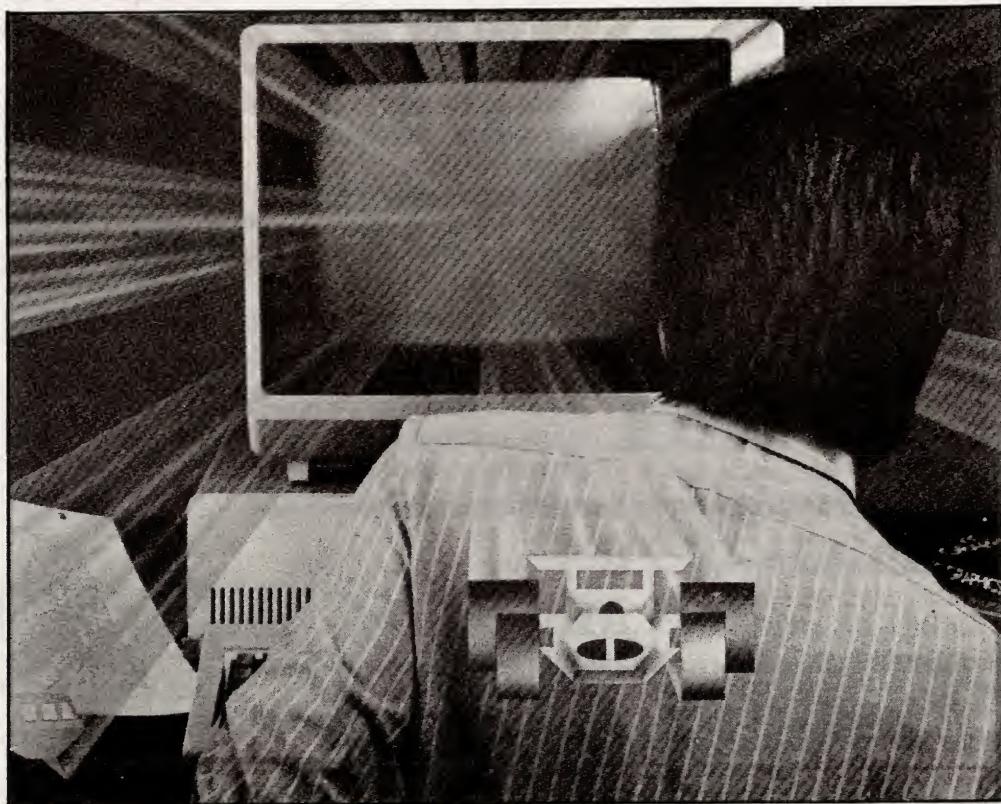
Hugo D. Caro

Es necesario agilizar la velocidad para que el software funcione de manera óptima. En la carga de juegos o en programas de comunicaciones, esta cualidad es imprescindible.

Si usted es un asiduo usuario del intérprete BASIC, tarde o temprano se habrá hecho esta pregunta: ¿Por qué es lento mi programa? La velocidad de ejecución de los programas es un tema clásico para los usuarios de microcomputadoras (como la MSX). El proceso que consiste en la medición de velocidad de un programa o sistema versus otro equivalente se lleva a cabo mediante programas que se denominan "benchmarks" o pruebas de escritorio. En la jerga informática de nuestro país se denomina "prueba de escritorio" aquella que se realiza con lápiz y papel, así que dejaremos el término yanqui "benchmark". Este proceso involucra la ejecución con tomas de tiempo de dos programas similares funcionalmente y la verificación de las diferencias de tiempo de corrida. Es una herramienta valiosa para comparar dos computadoras, pero no sirve para mucho más. En este artículo discutiremos diversas reglas que la práctica en el uso de nuestro bienamado intérprete BASIC nos ha dictado para acelerar el funcionamiento de los programas, y para verificar la diferencias de tiempo de ejecución usaremos un programita sencillo que iremos modificando hasta lograr una reducción interesante.

Los tiempos y ejemplos se han aplicado sobre el intérprete MSX-BASIC y sobre el MBASIC o BASIC 80 versión 5.21, que es la versión de BASIC que acompaña al compilador BASIC 80 disponible en plaza para nuestra norma. Sin embargo, aquel usuario que se tope con nuestras líneas podrá hacer uso de las reglas aquí brindadas ya que se pueden aplicar en general sobre cualquier intérprete BASIC.

Antes que nada quisiéramos aclarar que la velocidad de ejecución en un programa no siempre es crítica, y existen otros factores (por ejemplo, que sea de fácil lectura o que ahorre espacio de memoria) que son más importantes. Sin embargo, se dan situaciones donde realmente es necesario que la velocidad de ejecución sea el punto



de optimización del programa. Por ejemplo, los juegos tienen como problema principal tratar de reducir al máximo el tiempo de ejecución para lograr un efecto óptimo (es muy deprimente haber acertado un balazo al marcianito de siempre... y que la computadora no se dé por aludida). Asimismo, en los programas de comunicaciones (como los que traen el Modem y la Interfase RS-232 de Talent) también la velocidad es crítica, pues ésta es necesaria para que no se pierdan datos en el camino.

Antes de comenzar a sumergirnos en los mares de la velocidad de ejecución, veamos qué sucede cuando el usuario pulsa RUN si un programa sencillo, como por ejemplo el Programa 1, se encuentra en la memoria de nuestra Talent MSX:

Programa 1

```
10 LET N=0
15 REM *** COMIENZA EL CICLO
20 LET N=N+1
```

```
25 IF INT (N/7) = N/7
   THEN PRINT N;
30 REM ESTE CICLO VERIFICA A CADA NUMERO MENOR QUE 1000
35 REM PARA VER SI ES DIVISIBLE POR 7
40 REM SI ES ASI, SE IMPRIME EL NUMERO
45 IF N<1000 THEN GOTO 20
50 REM *** FIN DEL CICLO
```

La línea 10 está escrita en BASIC, por supuesto.

Muchos humanos pueden llegar a entender este lenguaje (si saben un poco de inglés), pero no es la lengua madre (lenguaje de máquina) de una microcomputadora. Antes de poder ejecutar la instrucción de la línea 10, la computadora debe poder obtener las instrucciones equivalentes en su lenguaje nativo. Más que investigar todos los detalles del BASIC interpretado, po-

demos imaginar la ejecución de una línea o sentencia BASIC como un proceso de dos pasos, a saber:

1. Traducir la sentencia BASIC a código de máquina.
2. Ejecutar la sentencia en código de máquina.

La mayor desventaja de las versiones interpretadas de BASIC es que el paso 1. consume demasiado tiempo. Algunas líneas, como por ejemplo, las que se encuentran dentro de ciclos, deben traducirse muchas veces cuando se ejecuta un programa. Veamos ahora que podemos hacer con nuestro Programa 1 para acelerarlo. El programa no cumple una tarea muy excitante que digamos -tan solo imprime los números menores que 1000 y además son múltiplos de 7-. Es un programa corto (excluyendo los REM's tiene sólo 4 líneas). Sin embargo, podremos ver que aun siendo un programa tan simple existen muchas técnicas que acelerarán considerablemente la ejecución del mismo.

Es bastante razonable suponer que nuestros esfuerzos para reducir el tiempo de ejecución deben centrarse sobre los ciclos, porque son los que más tiempo consumen cuando se ejecuta un programa. Esto nos sugiere que usemos nuestras lupas sobre las líneas 20 a 45.

Las sentencias REM dentro de este ciclo debe haberle llamado inmediatamente la atención. La mayoría de nosotros sabe (aún con cierta reticencia) que los REM's o comentarios son necesarios para una mayor legibilidad (vaya palabra) de un programa. Sin embargo, cuando se encuentran dentro de un ciclo, pueden retardar la ejecución de un programa en varios órdenes de magnitud. Si los comentarios deben estar incluidos en los listados definitivos de sus programas, asegúrense de que no se encuentren en el cuerpo de un ciclo. Atención: hemos encontrado nuestra primera "regla sagrada de la velocidad":

REGLA 1: ELIMINE TODOS LOS COMENTARIOS (REM'S) DE LOS CICLOS DEL PROGRAMA

Cuando aplicamos nuestra primera regla sacrosanta sobre el Programa 1, obtenemos la nueva versión del mismo, que llamaremos Programa 1.1. Este nuevo programa corre bastante más rápido que su predecesor, en el caso del MBASIC hasta un 28 % más rápidamente. Las sentencias REM de las líneas 15 y 45 no han sido eliminadas pues no ahorrarían mucho tiempo, al estar fuera del ciclo. Tampoco es necesario eliminar la inicialización de las variables, ya que no se verifica ninguna disminución significativa.

Programa 1.1

```
10 LET N=0
15 REM *** COMIENZA EL
  CICLO 20 LET N=N+1
30 IF INT (N/7) = N/7
  THEN PRINT N;
40 IF N<1000 THEN GOTO
  20
45 REM *** FIN DEL CI
  CLO
```

En general, ¿cuánto tiempo se ahorra realmente cuando se eliminan los REM's? Naturalmente, toda reducción de tiempo depende del número de comentarios que incluya el ciclo y las veces que se ejecuta el mismo. Pero también influye la longitud de las sentencias REM. No es lo mismo la "ejecución" de una sentencia REM con 3 caracteres que con 70...

En la mayoría de las versiones de BASIC, los ciclos pueden construirse de dos formas: con sentencias GOTO, y con sentencias FOR-NEXT (en el caso del MBASIC, también la sentencia WHILE permite, junto con WEND, crear ciclos).

En el Programa 1 hemos utilizado la sentencia GOTO, pero es obvio que se podría haber utilizado un ciclo FOR-NEXT de la siguiente manera:

Programa 1.2

```
20 FOR N=1 TO 1000
```

```
30 IF INT (N/7) = N/7
  THEN PRINT N;
```

```
40 NEXT N
```

Nótese que este programa es un poco más corto que el Programa 1 y que corre bastante más rápido. De hecho, en MSX-BASIC corre un 15% más rápido. Naturalmente, cuando se mejora la velocidad de ejecución de un programa se puede perder a veces un poco de la legibilidad del mismo. Sin embargo, en este caso no se aplica, ya que siempre se puede visualizar mejor un ciclo construido con FOR-NEXT que con GOTO. Llegamos pues a la segunda "regla sagrada":

REGLA 2: SIEMPRE QUE SEA POSIBLE, UTILICE CICLOS FOR-NEXT EN VEZ DE CICLOS CONSTRUIDOS CON SENTENCIAS GOTO.

Para ver más claramente la diferencia de tiempos, es mejor comparar la velocidad de ejecución de ciclos "vacíos". Esta medición tomará el tiempo de ejecución de la estructura en sí misma. Ciclo 1 y Ciclo 2 han sido tomados de los programas Programa 1 y Programa 2. La tabla 1 compara las velocidades de ejecución de ambos.

Ciclo 1

```
10 N=0
15 N=N+1
20 IF N<1000 GOTO 15
```

Ciclo 2

```
10 FOR N=1 TO 1000
20 NEXT N
```

TABLA 1:

Comparando ciclos vacíos

Programa	MBASIC	MSX-BASIC
Ciclo 1	6.44 s	6.00 s
Ciclo 2	2.44 s	2.25 s

Cuando se utilizan GOTO condicionales se obtiene más velocidad que cuando se utilizan los GOTO incondicionales. Por ejemplo, el siguiente ciclo utiliza un salto incondicional a la línea 30 y requiere un poco más de tiempo de ejecución. También puede aplicarse este criterio a los ciclos WHILE-WEND que existen en el MBA-



DEK

Soft

TODO EN CASSETTE
Y DISKETTE PARA
* MSX - COMMODORE
SPECTRUM - 2068

* FUNCIONAN EN TOSHIBA

VENTAS AL
POR MAYOR
Y MENOR
ENVIOS AL INTERIOR

AL MEJOR
PRECIO

ALSINA 1170 5º "511"

T.E. 37-3932/3954/0825/0891/4120 int. 511

SIC, pues los mismos corren un poco más lentamente que el Ciclo1, por lo tanto la Regla 2 también se aplica a los ciclos WHILE-WEND de la misma forma que con los ciclos creados con sentencias GOTO.

Ciclo 3

```
10 N=N+1
20 IF N>1000 GOTO 40
30 GOTO 10
40 REM EL PROGRAMA
CONTINUA AQUI...
```

Podemos aumentar aún más el tiempo de ejecución si eliminamos la N de la sentencia NEXT N, y así obtenemos al Ciclo 4. Cuando la sentencia NEXT se refiere a una variable, se demora el tiempo de ejecución.

Ciclo 4

```
10 FOR N=1 TO 1000
20 NEXT
```

Al no especificar la variable N en el Ciclo 4, eliminamos la necesidad de que la computadora busque una variable, y disminuye el tiempo de ejecución. Los ahorros son suficientes como para justificar el dictado de la tercera "regla sagrada":

REGLA 3: LOS CICLOS FOR-NEXT SE EJECUTAN MÁS RÁPIDAMENTE CUANDO NO SE ESPECIFICA LA VARIABLE EN LA SENTENCIA NEXT

Si aplicamos esta regla a nuestro Programa 1.2, obtenemos cerca de un 3% de aumento en la velocidad. Más adelante veremos que si aplicamos esta regla a un programa más complejo, el ahorro de tiempo será más significativo.

Programa 1.3

```
20 FOR N=1 TO 1000
30 IF INT (N/7) = N/7
THEN PRINT N;
40 NEXT
```

Hemos aplicado 3 reglas, y ahora nuestro programa sólo tiene 3 líneas de longitud. Sin embargo, ¡todavía pueden aplicarse dos reglas más!! La primera surge de la observación de que la división de la línea 30 se ejecuta dos veces. Es posible que el programa corra más rápidamente si encontramos una forma para que se ejecute sólo una vez. Pero esto es fácil: guardemos el resultado en una variable! Cuando hacemos esto, obtenemos el Programa 1.4.

Programa 1.4

```
20 FOR N=1 TO 1000
25 LET CHECK=N/7
30 IF INT (CHECK)=CHECK
THEN PRINT N;
40 NEXT
```

En MBASIC, el aumento de velocidad es despreciable, pero sobre MSX-BASIC

SIC se logró un aumento del orden del 27%. Sin embargo, el programa es ahora más largo y tiene una variable adicional; hemos canjeado velocidad por espacio de memoria. Este trueque se aplica reiteradamente cuando se necesita acelerar la velocidad de trabajo de un programa. Decidir cuándo debe aceptarse este canje no es siempre fácil, pero de todos modos tenemos nuestra siguiente regla sagrada:

REGLA 4: SI EL MISMO CALCULO SE EFECTUA MAS DE UNA VEZ, EL TIEMPO DE EJECUCION PUEDE ABREVIARSE A VECES UTILIZANDO UNA VARIABLE PARA ALMA-



CENAR EL RESULTADO Y LUEGO APLICARLA EN LA FORMULA, EN VEZ DE REPETIR EL CALCULO.

Al aplicar la Regla 4 al Programa 1.3, hemos podido eliminar un cálculo, N/7, a cambio de agregar una sentencia de asignación. Dado que la división es una operación que consume bastante tiempo en el MSX-BASIC, hemos obtenido un aumento significativo en el tiempo de ejecución. En cambio, en MBASIC, la división es un poco más eficiente y además se trabaja con números de simple precisión, y por lo tanto nuestro reemplazo no ahorra mucho tiempo.

La Regla 4 trabaja mejor cuando se eliminan más de dos cálculos. El reemplazo de una operación más rápida, como por ejemplo "+", debe hacerse con cuidado -puede ser necesaria cierta experimentación-. El tiempo de ejecución de un cálculo también depende de que sus operandos sean

constantes o variables. En algunas versiones de BASIC se puede ahorrar tiempo restringiendo el uso de constantes tales como 7 o 3.14159. Para implementar esta restricción podemos utilizar variables tales como S o PI. Luego, cada vez que se necesite utilizar una constante, se referencia mediante la variable apropiada. Esta es la idea que trasunta detrás de la Regla 5, que es la primera regla que no trabaja igual para MBASIC que para MSX-BASIC (en MBASIC no funciona):

REGLA 5: SI UTILIZA MUCHAS VECES UNA CONSTANTE DENTRO DE UN PROGRAMA, ENTONCES ALMACENE SU VALOR EN UNA VARIABLE Y LUEGO USE LA VARIABLE EN VEZ DE LA CONSTANTE.

Si aplicamos esta regla sobre el programa anterior, obtenemos:

Programa 1.5

```
10 LET S=7
20 FOR N=1 TO 1000
25 LET CHECK=N/S
30 IF INT (CHECK)=CHECK
THEN PRINT N;
40 NEXT
```

La mejora de velocidad, aun en MSX-BASIC, no alcanza al 4%. Sin embargo, los ahorros obtenidos con esta regla pueden ser substanciales. Por ejemplo, si la aplicamos a los siguientes segmentos de programa (ver Tabla 2), encontramos que la Regla 5 es efectiva aun para el MBASIC. Ambos segmentos calculan el área de un círculo 1000 veces.

Antes de la Regla 5

```
30 FOR N=1 TO 1000
40 LET AREA = 3.14159
* 14 * 14 50 NEXT
```

Después de la Regla 5

```
10 LET PI=3.14159
20 LET R=14
30 FOR N=1 TO 1000
40 LET AREA=PI*R*R
50 NEXT
```

TABLA 2

Tiempos de ejecución en segundos antes y después de la Regla 5.

	MBASIC	MSX-BASIC
Antes	7.12	12.41
Después	6.63	10.35

Existe una aplicación levemente diferente para la Regla 5, para las variables alfanuméricas. Cuando uno hace la asignación de una constante tal como CHR\$(12) por CLS\$, es mucho más rápido acceder a la variable CLS\$ que calcular CHR\$(12). Esta aplicación de la Regla 5 funciona también sobre el MBASIC.

Por último queda una regla más que se puede aplicar para ahorrar tiempo de ejecución. Esta regla es ligeramente diferente a las otras en la filosofía de

aplicación. Se pueden llegar a obtener ahorros espectaculares de tiempo, pero posiblemente sea muy difícil o virtualmente imposible de aplicar.

La regla es:

REGLA 6: SI LAS REGLAS ANTERIORES NO ACELERAN LO SUFICIENTE A SU PROGRAMA, TRATE CON OTRO ALGORITMO

En el caso de nuestro querido programa, la tarea a realizar se puede enfocar desde otro ángulo. La observación clave que se aplica aquí es que los números que son divisibles por 7 son precisamente aquellos que son múltiplos de 7 (por ejemplo, los números 7, 14, 21...). Podemos encontrar estos números por simple suma y evitar todas la verificación de números que sabemos que no son divisibles por 7. Generalmente, una nueva versión de un programa es significativamente más rápida que las versiones anteriores. El siguiente programa, por ejemplo, ¡¡corre al menos 23 segundos más rápidamente que la versión 1.1!!

Programa 1.6

```
10 LET S=7
15 REM *** COMIENZA EL CICLO
20 PRINT S;
30 LET S=S+7
40 IF S<1000 THEN 20
45 REM *** FIN DEL CICLO
```

En vista de las reglas que hemos analizado, es evidente que el programa 1.6 no es la versión más rápida que se puede implementar del nuevo algoritmo. Se puso este ejemplo para demostrar que un algoritmo superior es normalmente mucho mejor que cualquier versión anterior de un algoritmo inferior. El siguiente programa también está basado en esta nueva versión, pero se utiliza un ciclo FOR-NEXT con la opción STEP, que lo hace considerablemente más rápido que el ciclo

GOTO de antes:

Programa 1.7

```
10 FOR S=7 TO 1000
STEP 7
20 PRINT S;
30 NEXT
```

Es realmente sorprendente que se le haya podido aplicar 6 reglas de reducción de tiempos a un programa tan simple como Programa 1. La tabla 3 resume las mejoras de tiempo que se logran con las distintas versiones al aplicar las 6 reglas.

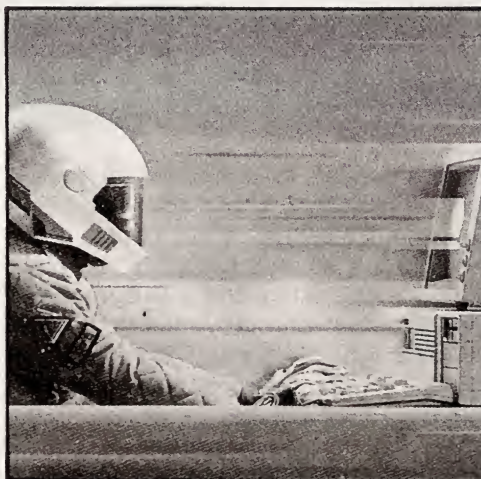


TABLA 3

Tiempos de ejecución después de aplicar reglas 1-6
(Ver nota al pie)

Regla	MBASIC	MSX-BASIC
Sin reglas	26.69	32.19
1	19.25	26.37
2	14.94	22.50
3	14.59	21.88
4	14.13	15.97
5	14.25	15.43
6	3.06	1.50
Programa 1.7	2.28	0.97

Aclaración: es notable la diferencia de velocidad a favor del MBASIC, pero se debe a varias circunstancias:

1. Como tipo por defecto utiliza variables simple precisión.

2. Se hicieron las pruebas sobre 80 columnas, lo que ahorra tiempo en la impresión.

Recordemos estas reglas cada vez que programemos, y encontraremos que se pueden podar varios segundos del tiempo de ejecución del programa. Continúen sintonizando esta estación (o sea compren los siguientes números de esta revista) ya que vendrán más chimentos sobre cómo acelerar la ejecución de sus programas.

NOTA IMPORTANTE:

En este artículo hemos asumido que Uds. desean escribir programas BASIC más eficientes que se ejecutarán sobre intérpretes. Una alternativa importante del BASIC interpretado es el BASIC compilado. Un compilador BASIC traduce el programa completo escrito en BASIC a código ejecutable de máquina antes de que el mismo se ejecute, y crea un módulo llamado de "run-time" o de tiempo de ejecución. Si se utiliza un compilador BASIC (como el que acompaña al MBASIC) la traducción de una línea de programa se efectúa una sola vez (durante la compilación del programa) para que cuando éste se ejecuta, no sea necesaria ninguna traducción adicional. Se deduce pues que un programa compilado ejecuta mucho más rápidamente que uno interpretado (el MSX-BASIC no posee aún un compilador como el BASIC 80, pero hay noticias que indican que aparecerán en algún momento...).

Si se requiere todavía más velocidad, se puede lograr creando programas eficientes en lenguaje Assembler o ensamblador. Sin embargo, para muchas personas es suficiente con los métodos que aquí se han explicado, ya que no tienen un compilador BASIC y no les interesa o no saben programar en lenguaje Assembler.

Hugo D. Caro

**ATENCION ! : LIBROS Y PROGRAMAS PARA
COMODORE - MSX - SPECTRUM
ATARI - AMSTRAD Y GENERALES.**

**DATA BECKER EL N.º 1 EN
INFORMATICA**

**OFERTA TODO SU CATALOGO A PRECIOS ESPECIALES
DIRECTAMENTE A TODOS LOS USUARIOS DE COMPUTADORAS**

PARAGUAY 783 P 11 "C" (1057) BS.AS. REP.ARGENTINA TEL:311-8632

DO RE MI

Clase: Entrenamiento musical
Autor: Manuel R. Rojas

(2da. Parte)

En el número anterior brindamos la presentación, descripción y partes de este programa. Además, en la lista de funciones llegamos hasta la elaboración de la segunda pantalla. En esta ocasión completamos el programa explicando el desarrollo del juego y cómo acceder a la tercera pantalla.

Con un disparo pasamos a la tercera pantalla, esta es la más simple. Elaborada sobre SCREEN 0, se utiliza para elegir el "Nivel" (entre 0 y 9) la cantidad de jugadores y la forma en que jugarán.

Una vez hecha la última elección (ellas pueden hacerse indistintamente desde el teclado o desde el joystick, fijándolas con un disparo) se pasa automáticamente a la cuarta y última pantalla. Aquí podemos visualizar el dibujo de un teclado completo de piano con dos cursores que se mueven independientemente y que sirven para acotar el intervalo de escala sobre el cual trabajaremos (se puede optar por cualquier longitud de "intervalo" ubicado a cualquier altura del teclado, aunque los cursores se presentan acotando una escala en la mitad del mismo). Debajo del "pianito" se ve un panel que marca dificultades para que nosotros elijamos su nivel:

1) "Intervalo" de escala musical (de este "intervalo" el programa extraerá al azar las "notas" que deberemos reconocer). Este se define por la nota de comienzo (NC%-cursor izquierdo) y la nota final (NF%-cursor derecho). Ambos se desplazan con movimientos izquierda/derecha y se fijan con un disparo.

2) Cantidad de "Tentativas" que creemos necesitar para acertar nuestra elección (CT%).

3) Cantidad de "Notas" que nos creemos capaces de reconocer (CN%). (Estas dificultades, que selecciona el propio jugador, definen un "Premio" (PM%) o puntaje máximo que podemos llegar a obtener en nuestra jugada).

4) "Volumen" del sonido (VS%).

Un ícono móvil indica al jugador qué opción está haciendo. Esta elección puede ser corregida todas las veces que se desee antes de comenzar a jugar, ya que la corrección del valor del "premio" es automática y, en cierta forma, otro factor más de elección, ya que la opción de juego puede ser alcanzado determinado puntaje por combinación de distintos tipos de dificultades (esto puede ser útil jugando de a dos, cuando se conocen las flaquezas del oponente o se especula con el or-



den de salida). El marcador de opciones se desplaza con arriba/abajo (siguiendo su propio movimiento). Las dificultades se aumentan/disminuyen "agregando" asteriscos a la derecha o "retirándolos" con movimiento a la izquierda. Los disparos fijan las decisiones y uno, sobre la orden de comienzo, inicia el juego en sí.

DESARROLLO DEL JUEGO

Primero se oirá la escala completa correspondiente al "intervalo" previamente determinado. Simultáneamente aparece un "marcador" que va señalando tecla a tecla (incluyendo bemoles y sostenidos) la ubicación de cada sonido. Terminada la escala este "marcador" salta al medio. Se escuchan entonces la o las "notas" que deberán reconocerse. Estas se repetirán durante el desarrollo del juego. Se comienza por adivinar la primera, y así sucesivamente, moviendo el "marcador" (izquierda/derecha) hacia la tecla que creemos que corresponde. Con un disparo fijamos nuestra decisión. Una señal auditiva indicará el acierto (campanillas) o el error (ruido grave). En el primer caso desaparecerá un asterisco de "notas" y se eliminará su sonido, dejando oír solo las restantes. En ambos casos desaparecerá un asterisco en "tentativas" y así disminuirán nuestras posibilidades de elección. De esta forma seguirá hasta adivinar todas las "notas" o gastar todas las "tentativas".

Durante el juego es posible reclamar una "ayuda" al programa, que consiste en la repetición de la emisión de la escala completa (incluyendo el "marcador" recorriendo el teclado). En este caso el "premio" se verá reducido en un factor de $0,1 \cdot NI\%$ por cada ayuda. Cuando el juego termina se otorga un puntaje valorativo del desempeño del

jugador o los jugadores. Cuando este sea máximo (y no se haya pedido ninguna "ayuda") el programa saludará el desempeño del usuario con una marcha.

El juego con dos jugadores es similar y lo ejecutan ambos simultáneamente. El acierto de uno, cuando se logra sin pedir ayuda, es premiado y con la posibilidad de hacer la jugada siguiente. El jugador que está en turno se indica en un recuadro debajo del pianito y aparecen ahora Premio 1 y 2, y Puntaje 1 y 2 (obviamente, la cuarta pantalla posee ligeras variaciones, según jueguen uno o dos jugadores).

LISTA DE FUNCIONES DEFINIDAS POR EL USUARIO

Pantalla 3

K%: Cursor (0...1).

J%: Renglón del cursor.

NI%: Nivel de juego (0...9).

JU%: Cantidad y equipamiento de jugadores (0...4).

I%: Compás del "Minué".

Pantalla 4

I%, J%, K%: Índices varios.

K%: Variable global: si $K\%=1$, el juego recién comienza; si no, ya se ha jugado alguna vez.

NC%: Nota de comienzo del intervalo.

NF%: Nota del final del intervalo.

CT%: Cantidad de tentativas.

CN%: Cantidad de notas a adivinar.

VS%: Volumen del sonido.

L%: STICK.

JU%: Cantidad y equipamiento de jugadores:

1 Un jugador con teclado.

2 Un jugador con joystick.

3 Dos jugadores con joystick (juega 1).

4 Dos jugadores (con joystick y teclado) juega 1.

5 Idem 3 pero juega 2

6 Idem 4 pero juega 2

A1%: Ayuda para el jugador 1.

A2%: Idem jugador 2.

NO%: Nota sobre el teclado.

CO%: Señalador de opción oscila entre 0 y 1.

PM%: Premio cuando juega un solo jugador.

LI%: Límite izquierdo del intervalo.

LD%: Idem derecho.

N%: Color de los apuntadores oscila entre 0 y 1.

M%: Modelo del SPRITE de los límites (-1 o 0).

P%: Posición del límite que se va corriendo.

J%: Dirección a la que apunta el STICK.

S% (15): Notas al azar (0...95).

CA%: Cantidad de aciertos para un solo jugador.

CE%: Cantidad de errores ídem.

PP%: Puntaje.

MX%: Puntaje máximo de la sesión.

P1%, P2%, P3%, P4%: Premios 1, 2, 3, 4 respectivamente.

E1%, E2%, E3%, E4%: Errores 1, 2, 3, 4 respectivamente.

AY%: Ayuda.

```
2480 '
2490 '
2500 '          PANTALLA 3
2510 '          Elección del nivel
2520 '
2530 COLOR15,4,4:SCREEN0:KEYOFF
2540 LOCATE5,3:PRINT"NIVEL (0..9): "
2550 LOCATE5,9:PRINT"1  UN JUGADOR CON TECLADO"
2560 LOCATE5,11:PRINT"2  UN JUGADOR CON JOYSTICK"
2570 LOCATE5,13:PRINT"3  DOS JUGADORES CON JOYSTICK"
2580 LOCATE5,15:PRINT"4  DOS JUGADORES CON JOY.Y TECL."
2590 K%=1:ONINTERVAL=760SUB2820:ONSTRIGGOSUB2840,2840,,2840,
2600 LOCATE19,3,K%:INTERVALON:STRIG(0)ON:STRIG(1)ON:STRIG(3)ON
2610 A$=INKEY$:IFA$="" THEN2630ELSEIFASC(A$)=13THENGOSUB2840
2620 IFASC(A$)>47ANDASC(A$)<58THENNI%=(1+NI%MOD10)ELSENIX%=(NI%+9)MOD10
2630 J%=STICK(0):IF0<J%THEN2640ELSEJ%=STICK(1):IF0=J%THEN2610
2640 IFJ%=10RJ%=20RJ%=3THENNI%=(1+NI%MOD10)ELSENIX%=(NI%+9)MOD10
2650 LOCATE18,3:PRINTNI%:LOCATE19,3,K%:FORJ%=0TO1000:NEXT:GOTO2610
2660 '
2670 '
2680 'Equipamiento del o los jugadores
2690 '
2700 '
2710 J%=9:K%=1:ONINTERVAL=1060SUB2850:ONSTRIGGOSUB2880,2880,,2880,
2720 STRIG(0)ON:STRIG(1)ON:STRIG(3)ON:INTERVALON
2730 LOCATE7,J,K%:FORL%=0TO2900:NEXT
2740 A$=STICK(0)+3:IF0=(A$MOD4)THEN2760ELSEA$=STICK(1)+3:IF0=(A$MOD4)THEN2760
2750 A$=INKEY$:IFA$="" THEN2740ELSEIF13=ASC(A$)THENGOSUB2880ELSEJU%=(ASC(A$)-48):IF
2760 JU%>40RJU%<1THEN2740ELSEJ%=7+2*JU%:GOTO2730
2770 '
2780 '
2790 '          Subrutinas para          interval y strig
2800 '
2810 '
2820 K%=1-K%:L%=PEEK(63833!)-PEEK(63834!):IF0<L%ANDL%<7THENGOSUB2870
2830 RETURN2600
2840 LOCATE18,3:PRINTNI%:INTERVALOFF:STRIG(0)OFF:STRIG(1)OFF:STRIG(3)OFF:RETURN2
710
2850 K%=1-K%:L%=PEEK(63833!)-PEEK(63834!):IF0<L%ANDL%<7THENGOSUB2870
2860 INTERVALON:LOCATE7,J,K%:RETURN2740
2870 I%=(I%+1)MOD40:PLAYMU$(0,I%),MU$(1,I%),MU$(2,I%):RETURN
2880 INTERVALOFF:STRIG(0)OFF:STRIG(1)OFF:STRIG(3)OFF:JU%=5*(J%-7):GOSUB2870:RET
URN2940
2890 '
2900 '
2910 '          PANTALLA 4
2920 '          Sprites para los apuntadores
2930 '
2940 SCREEN2:COLOR15,4:OPEN"grp:"FOROUTPUTAS#1:I%=$USR0(0)
2950 SPRITE$(0)=CHR$(8)+CHR$(8)+CHR$(8)+CHR$(20)+CHR$(34)+CHR$(34)+CHR$(20)+CHR$(
8)
2960 SPRITE$(1)=CHR$(8)+CHR$(8)+CHR$(8)+CHR$(24)+CHR$(24)+CHR$(24)+CHR$(56)+CHR$(
120)
2970 SPRITE$(2)=CHR$(8)+CHR$(8)+CHR$(8)+CHR$(12)+CHR$(12)+CHR$(12)+CHR$(14)+CHR$(
15)
2980 '
2990 '
3000 '          Pianito
```

CONCURSO

LOAD MSX

NOTAS, TRUCOS Y PROGRAMAS

AUSPICIADO POR TELEMATICA S.A. fabricante en Argentina de las computadoras personales TALENT MSX

Se premiarán los mejores trabajos. Los programas y trucos deberán servir para cualquier computadora de la norma MSX. Las notas deben apuntar a aprovechar los recursos de los equipos.

MENSUAL

Un programa provisto por Telemática, a elección entre IDEA BASE, IDEA TEXT o BASIC TUTOR



PREMIOS

TRIMESTRAL

Un software de aplicación, provisto por Telemática, con su manual pudiendo optar el ganador entre MSX LOGO, el MSX PLAN o el MSX WRITE.



Pueden escribir a nombre de CONCURSO MENSUAL LOAD MSX Paraná 720, 5º Piso (1017) Cap. Fed.

PROGRAMAS

```

3010 RINT#1,CHR$(210+CDX)
3640 M%=-1:N%=-1:PUTSPRITE1,(FNUX(CN%),40),15,1
3650 P%=-M%*(CN%-NF%)+NC%:LI%=-M%*(NC%+2):LD%=-M%*(NF%-97)+NF%-2
3660 INTERVALON:PUTSPRITE1-M%,(FNUX(P%),40),N%,1-M%
3670 IFL%THENSTRIG(L%):ON:STRIG(L%+2):ONELSESTRIG(O):ON
3680 J%=-STICK(L%):IFO=J%MOD2THEN3680
3690 FORJ%=0TO50:NEXTI%:IFL%THENSTRIG(L%):STOP:STRIG(L%+2):STOPELSESTRIG(O):STOP
3700 IF1=J%MOD4THEN3700ELSEJ%=-5*(5-J%)
3710 IF((J%+1)AND(M%ANDNF%LD%ORNOT(M%ANDNC%LD%))OR((J%=-1)AND(M%ANDNF%LI%ORNOT(M%ANDNC%LI%)))THENBEEP:GOTO3670
3720 NF%=-NF%-M%*J%:NC%=-NC%+(1+M%)*J%:IFK%THEN3650
3730 IFJUX<3THENGOSUB3570ELSEGOSUB3530:GOSUB5340
3740 LINE(0,80)-(39,87),12,BF:GOTO3650
3750 PUTSPRITE1-M%,(FNUX(P%),40),15,1-M%:M%=-NOT(M%):RETURN3650
3760 CD%=-NOT(CD%):LINE(16,72)-(23,79),12,BF:DRAW"bm16,72":PRINT#1,CHR$(210+CDX):
RETURN
3770 N%=-1-N%:PUTSPRITE1-M%,(FNUX(P%),40),N%,1-M%:RETURN
3780 INTERVALOFF:PUTSPRITE1-M%,(FNUX(P%),40),15,1-M%:FORI%=0TO4:STRIG(I%):OFF:NEX
TI%
3790 LINE(16,72)-(23,79),12,BF:IFK%THENRETURN3300
3800 IFJ%=1THENRETURN3390ELSERETURN3310
3810 ,
3820 ,
3830 'Cambiar la cantidad de tentativas
3840 ,
3850 ,
3860 ONINTERVAL=10GOSUB3950:INTERVALON:GOSUB3950
3870 DRAW"bm128,88":PRINT#1,STRING$(CT%,42):LINE(8*(16+CT%),88)-(8*(17+CT%),95),
12,BF:LINE(0,96)-(119,103),12,BF
3880 J%=-STICK(L%):IFO=J%MOD2THEN3880
3890 IFJ%=7ANDCT%=CN%ORJ%=3ANDCT%=15THENBEEP:GOTO3880
3900 IF3>J%MOD4THEN3900ELSECT%=CT%+5*(5-J%):IFK%THEN3870
3910 IFJUX<3THENGOSUB3570ELSEGOSUB5330:GOSUB5340
3920 GOTO3870
3930 INTERVALOFF:LINE(16,88)-(23,95),12,BF:IFK%THENRETURN3320
3940 IFJ%=1THENRETURN3290ELSERETURN3320
3950 CD%=-NOT(CD%):LINE(16,88)-(23,95),12,BF:DRAW"bm16,88":PRINT#1,CHR$(210+CDX):
RETURN
3960 ,
3970 ,
3980 'Cambiar la cantidad de notas
3990 ,
4000 ,
4010 ONINTERVAL=10GOSUB4100:INTERVALON:GOSUB4100
4020 DRAW"bm128,104":PRINT#1,STRING$(CN%,42):LINE(8*(16+CN%),104)-(8*(17+CN%),11
1),12,BF:LINE(0,112)-(119,119),12,BF
4030 J%=-STICK(L%):IFO=J%MOD2THEN4030
4040 IFJ%=7ANDCN%=10RJ%=3ANDCN%=CT%THENBEEP:GOTO4030
4050 IF3>J%MOD4THEN4050ELSECN%=CN%+5*(5-J%):IFK%THEN4020
4060 IFJUX<3THENGOSUB3570ELSEGOSUB5330:GOSUB5340
4070 GOTO4020
4080 INTERVALOFF:LINE(16,104)-(23,111),12,BF:IFK%THENRETURN3340
4090 IFJ%=1THENRETURN3310ELSERETURN3340
4100 CD%=-NOT(CD%):LINE(16,104)-(23,111),12,BF:DRAW"bm16,104":PRINT#1,CHR$(210+CD
%):RETURN
4110 ,
4120 ,
4130 ,
4140 ,
4150 ,
4160 ONINTERVAL=10GOSUB4230:INTERVALON:GOSUB4230
4170 DRAW"bm128,120":PRINT#1,STRING$(VS%,42):LINE(8*(16+VS%),120)-(8*(17+VS%),12
7),12,BF:LINE(0,128)-(119,135),12,BF
4180 J%=-STICK(L%):IFO=J%MOD2THEN4180
4190 IFJ%=7ANDVS%=10RJ%=3ANDVS%=15THENBEEP:GOTO4180
4200 IF3>J%MOD4THENVS%=VS%+5*(5-J%):GOTO4170
4210 INTERVALOFF:LINE(16,120)-(23,127),12,BF:IFK%THENRETURN3360
4220 IFJ%=1THENRETURN3330ELSERETURN3390
4230 CD%=-NOT(CD%):LINE(16,120)-(23,127),12,BF:DRAW"bm16,120":PRINT#1,CHR$(210+CD
%):RETURN
4240 ,
4250 ,

```


4260	Generaci "n aleatoria de notas	4310		4360		4410		4460		4510		4560		4610		4660		4710		4760		4810		4860		4910		4960		5000		5010		5020		5030		5040		5050		5060		5070		5080		5090		5100		5110		5120		5130		5140		5150		5160		5170		5180		5190		5200		5210		5220		5230		5240		5250		5260		5270		5280		5290		5300		5310		5320		5330		5340		5350		5360		5370		5380		5390		5400		5410		5420		5430		5440		5450		5460		5470		5480		5490		5500		5510		5520		5530		5540		5550		5560		5570		5580		5590		5600		5610		5620		5630		5640		5650		5660		5670		5680		5690		5700		5710		5720		5730		5740		5750		5760		5770		5780		5790		5800		5810		5820		5830		5840		5850		5860		5870		5880		5890		5900		5910		5920		5930		5940		5950		5960		5970		5980		5990		6000		6010		6020		6030		6040		6050		6060		6070		6080		6090		6100		6110		6120		6130		6140		6150		6160		6170		6180		6190		6200		6210		6220		6230		6240		6250		6260		6270		6280		6290		6300		6310		6320		6330		6340		6350		6360		6370		6380		6390		6400		6410		6420		6430		6440		6450		6460		6470		6480		6490		6500		6510		6520		6530		6540		6550		6560		6570		6580		6590		6600		6610		6620		6630		6640		6650		6660		6670		6680		6690		6700		6710		6720		6730		6740		6750		6760		6770		6780		6790		6800		6810		6820		6830		6840		6850		6860		6870		6880		6890		6900		6910		6920		6930		6940		6950		6960		6970		6980		6990		7000		7010		7020		7030		7040		7050		7060		7070		7080		7090		7100		7110		7120		7130		7140		7150		7160		7170		7180		7190		7200		7210		7220		7230		7240		7250		7260		7270		7280		7290		7300		7310		7320		7330		7340		7350		7360		7370		7380		7390		7400		7410		7420		7430		7440		7450		7460		7470		7480		7490		7500		7510		7520		7530		7540		7550		7560		7570		7580		7590		7600		7610		7620		7630		7640		7650		7660		7670		7680		7690		7700		7710		7720		7730		7740		7750		7760		7770		7780		7790		7800		7810		7820		7830		7840		7850		7860		7870		7880		7890		7900		7910		7920		7930		7940		7950		7960		7970		7980
------	--------------------------------	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------

EL BASIC MSX Y EL DE PC

Nos referimos a las instrucciones que son iguales o de resultado similar, que permiten usar un programa de MSX en una PC.



INSTRUCCION	MSX BASIC	BASICA	COMENTARIOS
ABS(n)	SI	SI	
ASC (n\$)	SI	SI	
ATN(r)	SI	SI	En radianes
AUTO com, inc	SI	SI	
BEEP	SI	SI	
BLOAD "nom"	SI	SI	
BSAVE "nom" (x)	NO	NO	No son compatibles
CALL rutina (n) *	SI	SI	Teniendo en cuenta rutina
CDBL (n)	SI	SI	
CHR\$ (n)	SI	SI	
CINT (n)	SI	SI	
CIRCLE (x, y), r, c, a, b, e	SI	SI	Verificando la resolución
CLEAR var, rntop	SI	SI	
CLOSE n1, n2,...	SI	SI	
CLS	SI	SI	
COLOR t, f, b	SI	SI	En PC modo texto
CONT	SI	SI	
COS (n)	SI	SI	En radianes
CSNG (n)	SI	SI	
CSRLIN (n)	SI	SI	Equivalente en PC a POS (n)
CVD (n\$)	SI	SI	
CVI (n\$)	SI	SI	
CVS (n\$)	SI	SI	
DATA n, m, p, ...	SI	SI	
DEF USR = d.rut	SI	SI	
DELETE c, f	SI	SI	
DIM a (m, n), b (p),...	SI	SI	Equivalente en PC a DIMTABL
EDIT 1	SI	SI	
ELSE	SI	SI	
END	SI	SI	
EOF	SI	SI	
ERASE u, v	SI	SI	
ERL	SI	SI	
ERR	SI	SI	
ERROR	SI	SI	
EXP (n)	SI	SI	
FIELD #n, l1 AS p\$	SI	SI	
FILES	SI	SI	
FIX (n)	SI	SI	
FOR TO STEP	SI	SI	
FRE (0)	SI	SI	
FRE (" ")	SI	SI	
GET # c, n	SI	SI	
GOSUB II	SI	SI	
GO TO II	SI	SI	
HEX\$ (n)	SI	SI	
IF THEN ELSE	SI	SI	
INKEY\$	SI	SI	
IMP (p)	SI	SI	
INPUT a, b, c,...	SI	SI	En PC se llama INP (P)
INPUT #c,a,b,c,...	SI	SI	
INPUT\$ (n)	SI	SI	
INSTR (cad, arg)	SI	SI	
INT (n)	SI	SI	
KEY ON	SI	SI	
KEY OFF	SI	SI	
KEY n, "mens"	SI	SI	
ON KEY GOSUB II	SI	SI	
KEY (n) ON, OFF, STOP	SI	SI	
KILL "nom"	SI	SI	
LEFT\$(cad, lon)	SI	SI	
LEN (a\$)	SI	SI	
LET	SI	SI	Ya no se emplea
LINE (x, y)-(u, v), c, op	SI	SI	
LINE INPUT "mens"; a\$	SI	SI	
LINE INPUT # c, a\$	SI	SI	

Mucho se ha dicho acerca de la compatibilidad de la norma MSX respecto del formato de disco, del CP/M, de los caracteres, de los archivos, etcétera. Estas ventajas permiten acercarse a terrenos más profesionales. Es indudable la satisfacción que se siente al encontrar una máquina que le permita a uno insertar un disco de una PC y poder leerlo. Entonces uno, ya más tranquilo, enciende su computadora y se dispone a hacer un programa en BASIC y, una vez concluido se deleita con el "nene" recién nacido. Es en este momento cuando llega un amigo que casualmente necesita un programa que hace las mismas cosas que el nuestro pero para una PC; surge ahí el dilema del BASIC. ¿Será también compatible? ¿Hacemos el programa desde cero en una PC, o usamos el nuestro?

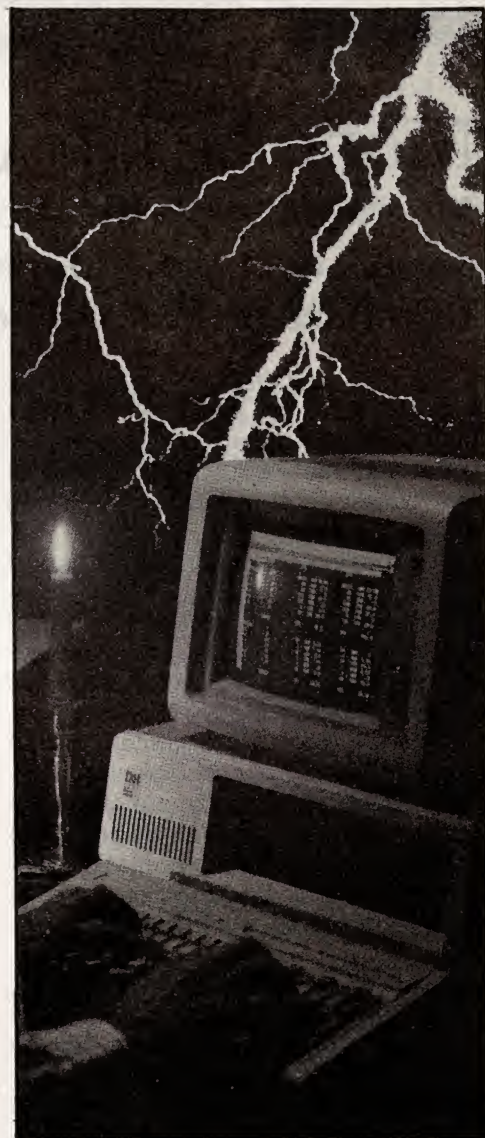
Por eso para clarificar el panorama analizaremos el MSX-BASIC y el BASIC de IBM llamado BASICA pero nos referiremos solamente a las instrucciones que son iguales o de resultado similar:

CONSTANTES

Se definen de igual manera en los dos BASIC: DEFDBL, DEFSNG, DEFINT y DEFSTR.

Los tipos de base que podemos usar

LIST com, fin	SI	SI	También en sus variantes
LLIST com, fin	SI	SI	Idem
LFILES	SI	SI	
LOAD "nom"	SI	SI	Y en su versión, R
LOC (c)	SI	SI	
LOCATE I, c	SI	SI	En PC se agregan comandos
LOF (c)	SI	SI	
LOG (n)	SI	SI	
LPOS (x)	SI	SI	En PC x vale 1, 2 o 3
LPRINT USING	SI	SI	En todos sus formatos
LSET camp\$, a\$	SI	SI	
MERGE	SI	SI	
MID\$ (cad\$, pos, lon)	SI	SI	
MKD\$ (dbl)	SI	SI	
MKIS (ent)	SI	SI	
MKS\$ (sim)	SI	SI	
MOD	SI	SI	
MOTOR	SI	SI	
NAME "vie" AS "nue"	SI	SI	
NEW	SI	SI	
NEXT a, b,...	SI	SI	
OCT\$ (n)	SI	SI	
ON ERROR GOTO (GOSUB)	SI	SI	
OPEN (*)	NO	NO	Difiere la programación
ON STOP GOTO GOSUB	SI	SI	Equivalencia en PC a ON STRING
ON STRIG GOTO GOSUB	SI	NO	
STOP ON	SI	NO	
ON INTERVAL	SI	NO	
OUT p, dat	SI	SI	
PAINT (x, y), c	SI	SI	En PC se agregan col y lim
PEEK (dir)	SI	SI	
POINT (x, y)	SI	SI	
POKE dir, dat	SI	SI	
POS (0)	SI	SI	
PRESET (x, y)	SI	SI	
PRINT a, b\$,...	SI	SI	En todos sus formatos
PRINT # c, a, b,...	SI	SI	
PSET (x, y), c	SI	SI	
PUT #c, n	SI	SI	
READ a, b\$, C,...	SI	SI	
REM	SI	SI	En todos sus formatos
RESTORE II	SI	SI	
RESUME	SI	SI	
RETURN II	SI	SI	
RIGHT\$ (cad\$, lon)	SI	SI	
RND (+-n)	SI	SI	
RSET cap\$, a\$	SI	SI	
RUN II	SI	SI	
RUN "nom"	SI	SI	
SAVE "NOM"	SI	SI	En todos sus formatos
SCREEN n, (x)	SI	SI	Compatible solo en n
SGN (n)	SI	SI	
SIN (n)	SI	SI	
SPACES\$ (n)	SI	SI	
SPC (c)	SI	SI	
SQR (n)	SI	SI	
STEP (x, y)	SI	SI	
STICK (n)	SI	SI	
STOP	SI	SI	
STR\$ (n)	SI	SI	
STRIG ON/OFF/STOP	SI	SI	
STRIG (n)	SI	SI	
STRING\$ (n, c\$)	SI	SI	En todos sus formatos
SWAP n, m	SI	SI	
SYSTEM	SI	SI	En MSX agregar call o _
TAB (col)	SI	SI	
TAN (n)	SI	SI	
TIME	SI	NO	Difiere el formato
TROFF	SI	SI	
TRON	SI	SI	
USR n (x)	SI	SI	
VAL (n\$)	SI	SI	
VARPTR (n)	SI	SI	En todos sus formatos
WAIT p, msq1, msq2	SI	SI	
WIDTH col	SI	SI	



son: &H para la hexadecimal, &O para la octal, &B para la binaria (MSX solamente).

VARIABLES

Tienen cuatro variantes: enteras (%), simple precisión (!), doble precisión (#) y cadenas o strings (\$).

OPERADORES

Ambos soportan: +, -, *, /, ^.

OPERADORES RELACIONALES

Idem: =, <>, <, >, <=, >=.

OPERADORES LOGICOS

Idem: NOT, AND, OR, XOR, IMP, EQV.

FUNCIONES PERSONALIZADAS

Se mantiene la estructura DEF FN nombre=expresión: FN nom.

No hemos incluido ni DRAW ni SOUND ni PLAY ya que su tratamiento como macro no las hacen totalmente compatibles con PC pero hablaremos de ellas más adelante. La apertura de canal "OPEN" en BASICA tiene los siguientes formatos: OPEN TIPO\$, #CANAL, NOMBRE\$, LONG. TIPO\$ toma el valor de "I", "O", "R" o "A" según sea lectura secuencial, escritura secuencial, agrega al final o acceso directo; solo para el último se agrega la longitud de registro. Si modificamos esta instrucción en nuestro programa, podremos hacerlo ejecutar sin mayores problemas.

MODO DE ENCARAR LA ENSEÑANZA DEL LOGO

Mucho se habla de este lenguaje y su aplicación en la formación informática del niño. El Licenciado Pablo A. Ger, director del Instituto Nueva Enseñanza de Vicente López clarificó el panorama.

¿Por qué sí LOGO?

LOGO es un lenguaje para programar computadoras, estructurado y simple, que posee un contenido matemático adaptable a los niños.

Es un medio matemáticamente expresivo

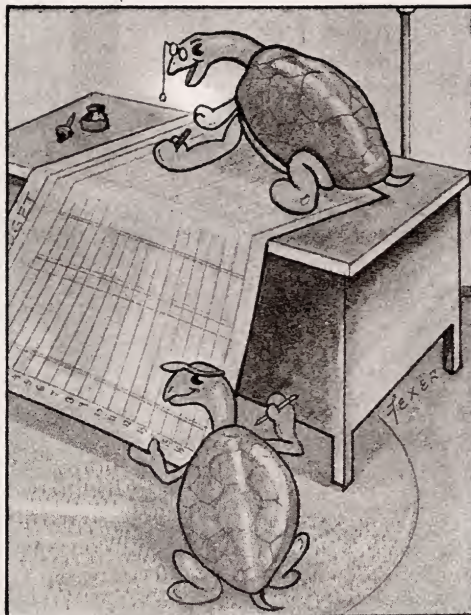
Y éste en muchos sentidos (el espacio, el movimiento, las pautas de acción y repetición) no llega a los niños a través de los métodos tradicionales. Todas estas características hacen que a través de la enseñanza del LOGO, el niño no solamente pueda interactuar con una computadora, sino también aprenda las matemáticas como el conjunto de principios orientadores que rigen el aprendizaje.

¿Por qué no LOGO?

En la enseñanza de un lenguaje de programación de computadoras, cualquiera sea, se enfatiza el aprendizaje de éste y luego el adquirir distintos tipos de conocimientos, pero siempre como algo secundario.

Son dos caminos distintos, si bien se unen en un punto. El que se sigue para enseñar LOGO es mucho más corto y

menos trabajoso, pero con una visión de futuro menos amplia y poco duradera.



¿Y entonces qué?

Podemos ver que lo que en definitiva le queda al niño es el conjunto de conductas que pudo aprender a través del

estudio del lenguaje y no el LOGO en sí mismo, como un conjunto de órdenes que si bien son fáciles de aprender también son fáciles de olvidar.

El entrenamiento de las mencionadas conductas no excluyen el aprendizaje del lenguaje. Por el contrario a través de él y su práctica, el niño las interioriza, haciéndolas propias, y lo beneficiará no sólo por haber aprendido un medio matemático expresivo, sino como el medio para descubrir, comprender y expresar su propio proceso de pensamiento.

¿Para qué el LOGO y la computadora?

Es indudable que al introducir una computadora en el aula, se revolucionan los métodos tradicionales de enseñanza.

El maestro ahora adquiere la tarea de diseñar formas ingeniosas de motivar a los niños para que aprenda cosas que naturalmente no aprendía. El docente asume el papel de guía y el alumno, el de creador y dueño absoluto de un conocimiento que le es propio. Usando como herramientas del aprendizaje una computadora y su lenguaje,

CENTROS DE ASISTENCIA AL USUARIO DE TALENT MSX

CAPITAL FEDERAL

Centro Cultural de la Ciudad de Buenos Aires

Taller Logo de computación
Junín 1930
Martes a Sábados de 15 a 19.30 horas

Fundación de Informática y Educación

Centro de Computación Clínica
Asistencia al Usuario Discapacitado
Ramsay 2250 - Pabellón F
Tel. 784-2018
Lunes a Viernes de 8 a 17 horas

Barrio Norte

Uriburu 1063 - Tel. 83-6892/826-

6692

Lunes a Viernes de 9 a 21 horas
Sábados de 9 a 12 horas

Belgrano

Mendoza 2728 - Tel. 781-2271
Lunes a Viernes de 15 a 22 horas

Centro

Av. Córdoba 654 - Tel. 392-5328/7611/8043/8051/8251
Lunes a Viernes de 12 a 21 horas
Sábados de 9 a 13 horas

Flores

Gral. Artigas 354 - Tel. 612-3902
Lunes a Viernes de 14 a 20 horas
Sábados de 10 a 13 horas

Palermo

Guatemala 4733 - Tel. 71-4124
Lunes a Viernes de 14 a 21 horas
Sábados de 9 a 13 horas

San Telmo

Chile 1345 - Tel. 37-0051 al 54
Lunes a Viernes de 10 a 13 y de 14 a 19 horas

GRAN BUENOS AIRES

Lanús

Caaguazú 2186 - Tel. 247-0678
Lunes a Viernes de 9 a 13 y de 16 a 20 horas
Sábados de 9 a 13 horas

Morón

Belgrano 160 - Tel. 629-3347
Lunes a Viernes de 9 a 13 y de 14 a

se logra que el niño adquiera una forma de pensar, de razonar, que lo beneficia para poder atrapar sus ideas, expresarlas y comprender los conocimientos que le brindará la escuela, no para colocarlos en un recipiente de ideas, sino encontrándoles un fundamento teórico-práctico.

Todo esto hace que el niño obtenga un aprendizaje natural, el que tiene lugar cuando alguien está en contacto con los materiales adecuados, y que a través de un juego no competitivo llegue a usar la computadora para el desarrollo intelectual.

¿Hasta dónde puede llegar el niño?

A medida que el niño avanza en el conocimiento de su pensamiento y va tomando sus ideas, a través de un entrenamiento que le hace cada vez más fácil comprenderlas, se introduce en un mundo de creatividad absoluta, al ver que sus ideas se transforman en una realidad expresada en la pantalla de la computadora. Aunque no llegue a lograr totalmente lo que pretende, sus errores producen una retroalimentación que le permite aprender sobre ellos.

Para poder avanzar, él mismo va a ir solicitando nuevas herramientas del lenguaje que harán más simple o más sofisticado su trabajo. Esto implica que puede avanzar tanto como su imaginación se lo permita. En la mayoría de los casos, al llegar a cierta edad, el niño solo va dejando su lenguaje de trabajo (en este caso el LOGO) para requerir otro.

Todo lo mencionado lleva a la conclusión de que es el niño el que pone sus límites y no un programa pre-establecido.

¿Se producen cambios en el niño?

Si, se produce un cambio que varía de chico en chico. Existen niños que naturalmente hacen un deporte en forma asombrosa o toman un pincel y dibujan, como artistas. Otros sabrán pensar de una forma innata en ellos, resolverán problemas matemáticos con felicidad o estudiarán con rapidez. A unos les dará la capacidad de reflexión y comprensión, útiles para el aprendizaje, les permitirá la toma de ciertos conocimientos a través de la reflexión conciente, y a aquellos cuya capacidad de reflexión es innata les ayudará a ordenar sus pensamientos con rapidez y eficiencia.

¿Por qué la división por edades y niveles?

A lo largo de la escuela primaria el niño va adquiriendo distintos niveles de captación y abstracción, por ello los grupos de trabajo se dividen por edades y niveles, para lograr de esta manera una homogeneidad de aprendizaje.

Tanto en la escuela primaria como secundaria, los cambios de un año al siguiente son importantes, en cuanto a cantidad y calidad de conocimientos. Cada uno de los cursos, además de estar dividido por edades y niveles de captación, está caracterizado por un

logro final a obtener a través de distintas conductas. Como producto final realizará un hito, un proyecto personal de absoluta creatividad donde procurará volcar sus ideas, corregirlas, perfeccionarlas, aprender con sus errores y hasta mostrar sus trabajos, por ejemplo, a sus padres logrando su participación y cooperación.

CONCURSO MENSUAL

Se realizó el primer sorteo de cupones auspiciado por Telemática y los Centros de Asistencia al Usuario. El **primer premio** (una computadora Talent MSX DPC-200) lo ganó **Bárbara Vázquez**, de Florida. El **segundo premio** (un kit educativo cada uno), **Carlos Leeb** de Valentín Alsina, **Laura de Calaveia** de Capital y **Karina Frago** de Lanús. El **tercer premio** (una beca cada uno para un curso en el Centro de Asistencia a elección), **Agustín Rosato** de Olivos, **María Concepción Barriónuevo** de Chilavert, **Miguel Holli-day** de San Isidro, **Carolina Sarquis** de Capital, y **Nelly Rosa Bruny** de El Palomar.

Este es el primero de una serie de sorteos mensuales. Para todos habrá los mismos premios. Quienes deseen participar deberán dirigirse a cualquier una de las direcciones de los Centros de Asistencia para retirar los cupones y recibir las indicaciones de cómo llenarlos.

21 horas

Sábados de 9 a 13 horas

Ramos Mejía

Bolívar 55 - 1er. piso - Tel. 658-4777

Lunes a Viernes de 9 a 13 y de 14 a 21 horas

Sábados de 9 a 13 horas

San Isidro

Av. Centenario 705 - Tel. 743-9678/747-6094

Lunes a Viernes de 9 a 21 horas

Sábados de 9 a 12 horas

Vicente López

Av. Maipú 625 - Tel. 797-6720

Lunes a Viernes de 10 a 19 horas

INTERIOR DEL PAIS

La Plata - Pcia.de Buenos

Aires

Calle 48 No. 529 - Tel. (021) 249905 al 07

Lunes a Viernes de 9 a 21 horas

Sábados de 9 a 13 horas

Bahía Blanca - Pcia.de

Buenos Aires

Gral. Paz 257 - Tel. (091) 31582

Lunes a Viernes de 9 a 12 y de 16 a 20 horas

Córdoba - Pcia.de Córdoba

9 de julio 533

Lunes a Viernes de 8 a 12 y de 16 a 20 horas

Villa María - Pcia.de Córdoba

Corrientes 1159 - 2do. piso - Tel (0535) 24311

Lunes a Viernes de 16 a 23 horas

Sábados de 8 a 12 y de 15 a 18

Mendoza - Pcia.de Mendoza

Rivadavia 76 - 1er. piso - Tel. (061)

291348/293151

Lunes a Viernes de 8 a 13 y de 16 a 20 horas

Sábados de 8 a 13 horas

Santa Fe - Pcia.de Santa Fe

Rivadavia 2553 Loc.22 - (042) 41832

Lunes a Viernes de 9 a 12 y de 16 a 19 horas

Sábados de 9 a 12 horas

Rosario - Pcia.de Santa Fe

Barón de Mau 1088

Lunes a Viernes de 8 a 12 y de 15 a 19 horas

Sábados de 9 a 12 horas

S.M.de Tucumán - Pcia.de Tucumán

Bolívar 374 - Tel. (081) 245007

Lunes a Viernes de 9 a 18 horas

SCRASH

AUTOR: JORGE GUILLERMO TENTOR

CLASE: ENTRETENIMIENTO



También es llamado juego de los números (se reemplazan los colores por números).

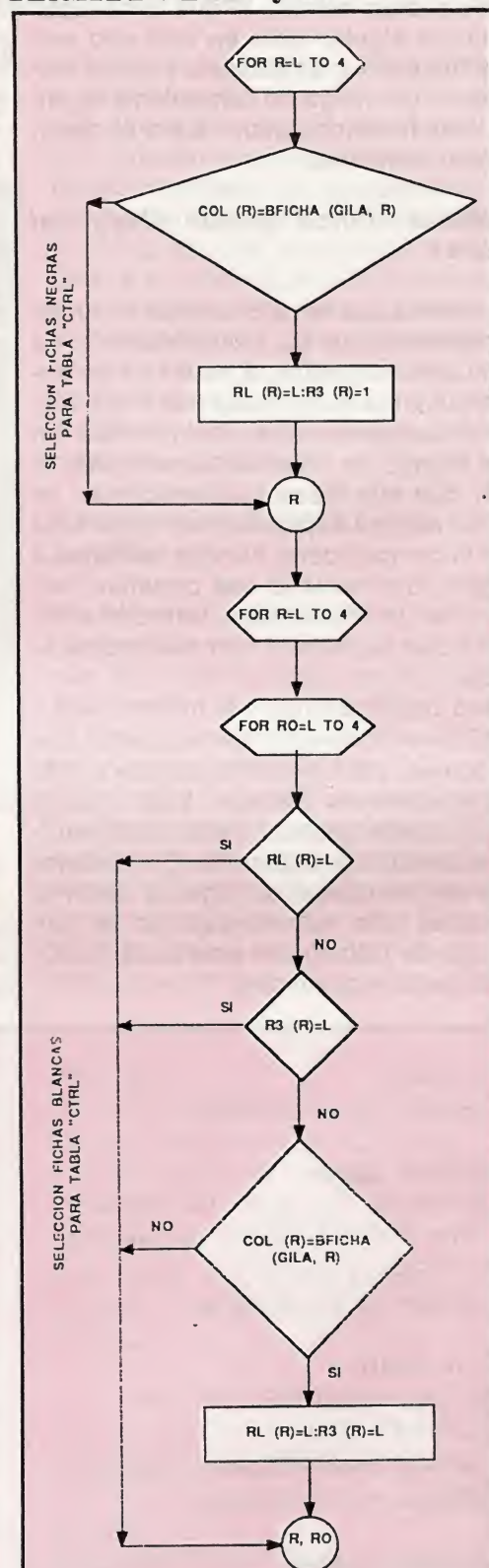
Este juego consiste en adivinar la combinación secreta de los colores elegidos por la computadora. El número

de fichas es de cuatro (4) y el de los colores es de cinco (5) (blanco, verde, rojo, amarillo y negro).

Las fichas de colores se eligen mediante las teclas de función. Luego de que el jugador elige las cuatro fichas, la computadora nos indica cuántas coin-

ciden en color y posición, ubicando en la tecla "CTRL" las correspondientes fichas negras. Luego el programa nos indica con las blancas la cantidad de fichas (del jugador) que coinciden en color, pero no en posición con las fichas del código secreto.

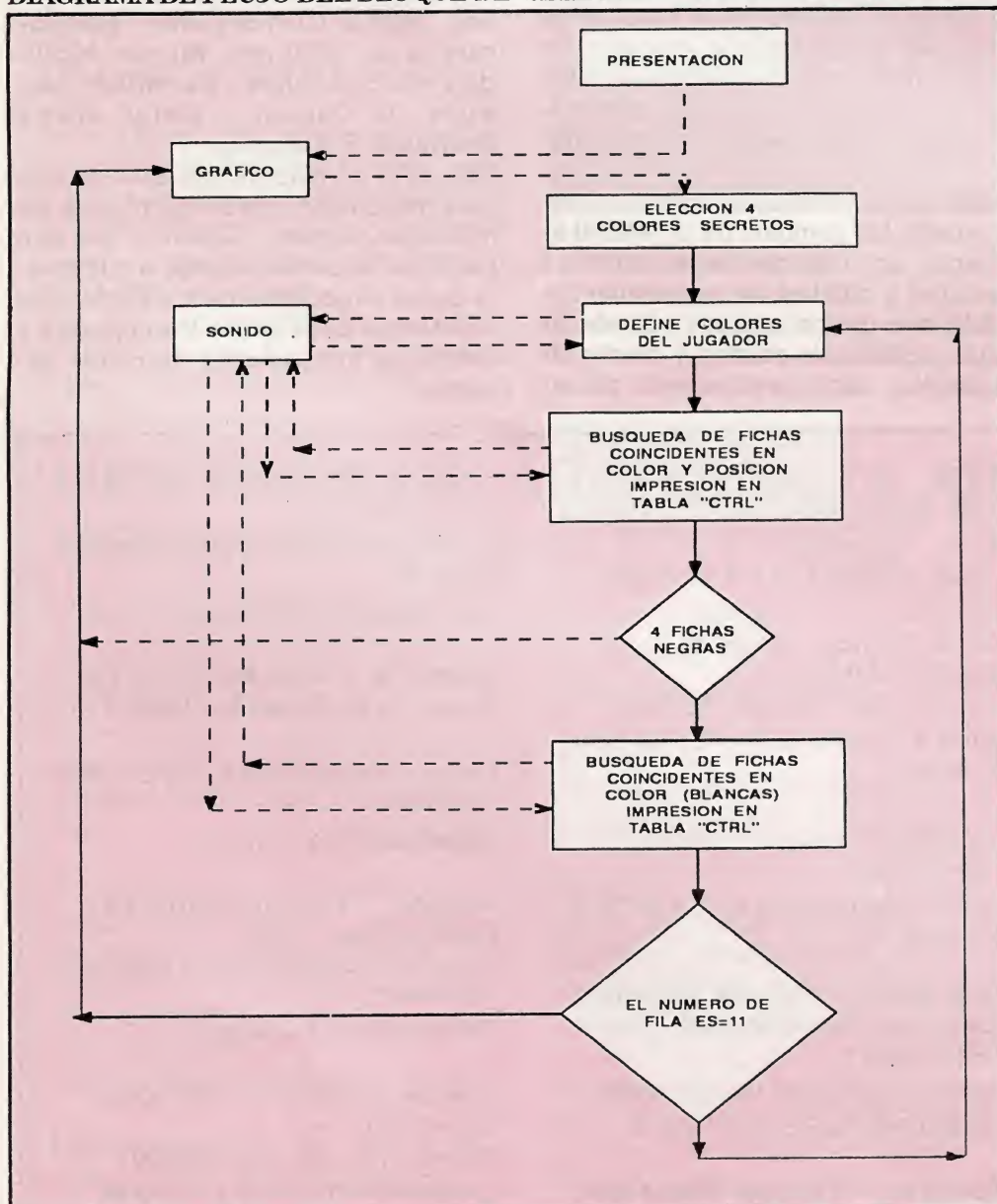
DIAGRAMA DE BLOQUES



Este diagrama es necesario para la aclaración de cómo el otro jugador (la computadora) nos indica cuántas fichas negras nos corresponden y cuántas fichas blancas.

Este diagrama corresponde al bloque de líneas 570-790.

DIAGRAMA DE FLUJO DEL BLOQUE DE SELECCION DE FICHAS PARA CTRL



Es conveniente destacar que las fichas registradas con puntos negros en la tabla "CTRL" no vuelven a ser registradas para los puntos blancos en la tabla de control ("CTRL"). Se pueden repetir los colores.

Ejemplo:

rojo verde negro rojo -----

Código secreto

rojo blanco verde amarillo-----

Primera fila

En este caso la tecla "CTRL" nos indicará con un punto negro (rojo) y un punto blanco (verde).

Al seguir jugando pasamos a la segunda fila.

negro verde blanco rojo-----

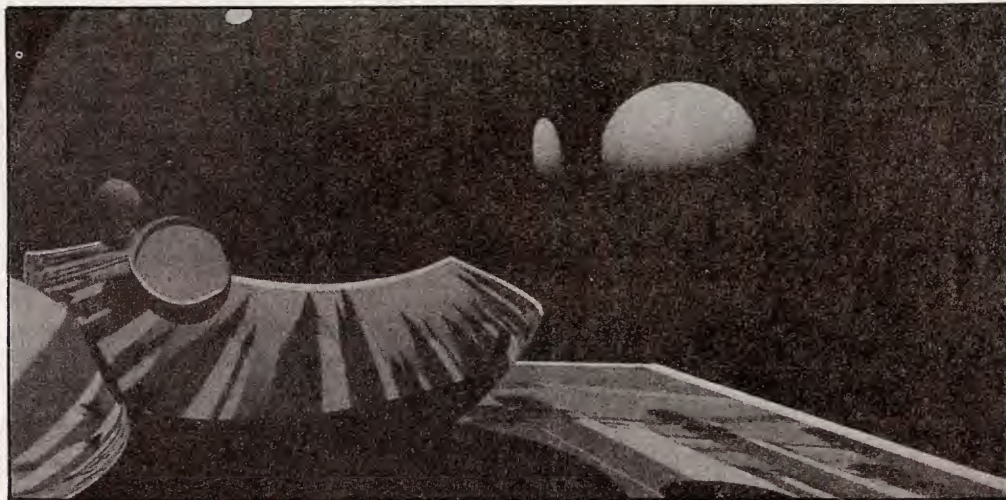
Segunda fila

En este caso la tabla "CTRL" nos indicará con dos puntos negros (verde-rojo) y un punto blanco (negro).

El juego termina cuando se descubre el código secreto o bien cuando llegamos hasta la última fila (11), en ambos casos se muestra el código secreto y se debe teclear "1" para seguir jugando

VARIABLES IMPORTANTES

FILA: Coordenada "y" para gráficos (fichas).



XCOL: Coordenada "x" para gráficos (fichas).

GILA: Fila en juego.

X: Columna en juego.

KOL: Color elegido por el jugador.

BFICHA (,): Matriz con los colores de las fichas.

ZCOL: Coordenada "x" para fichas de tabla "CTRL".

CTRL: Número de fichas acertadas en color y posición (negras).

COL(): Color de las cuatro fichas secretas.

R1(): Ficha del código secreto.=1 si es coincidente con la ficha del jugador.

R3(): Ficha del jugador.=1 si es coincidente con la ficha del código secreto.

BLOQUES POR NUMEROS DE LINEA

10-110: presentación "SCRASH".

120: Ciclo de retardo.

200-280: Elige 4 colores secretos.

320-450: Impresión fichas del jugador.

460-470: Inicializa variables.

570-630: Registra fichas negras.

700-790: Registra fichas blancas.

MICROBYTE Software

Precios al 5/10/87

MONTEVIDEO 252 (1019) CAP. Tel: 38-0331

SERVICIO PUERTA A PUERTA DE CASSETTES MSX

081-ATHLETIC LAND	A10
0802-HYPER SPORT 1	A10
0803-HYPER SPORT 2	A10
0806-ROAD FIGHTER	A10
0809-FLIGHT PATH 737	A10
0813-SUPER CHESS	A10
0814-YIE AR KUNG FU	A10
0826-KONAMI'S SOCCER	A10
0827-BOULDER DASH	A10
0837-RIVER RAID	A10
0843-GOLF (HOLE IN ONE)	A10
0844-SAMURAI NINJA	A10
0847-TENNIS KONAMI	A10
0850-ANTARCTIC ADVENTURE	A10
0856-F-16	A10
0864-SKY JEGUAR	A10
0873-FROGGER	A10
0874-PINGUILANDIA	A10
0875-CHOPFLITER	A10
0876-GALAGA	A10
0877-KUNG FU MASTER	A10
0881-FRUIT PANIC	A10
0882-EXERION	A10
0884-SUPER SINTH	A10
0892-GHOSTBUSTERS	A10
0898-PAC-MAN	A10
0903-PENGUIN'S WAR	A10
0904-EGGERLAND MISTERY	A10
0905-EXERION II	A10
0908PIPPOLS	A10
0909-COSMO EXPLORER	A10
0910-SKI COMMAND	A10
0917-B.C.QUEST	A10
0918-CHAPION PRO WRESTLING	A10
0919-VOLGUARD	A10

0920-THE DAM BUSTERS	A10
0922-GRAND PRIX	A10
0923-COSMOS	A10
0924-PING-PONG	A10
0925-RED ZONE	A10
0928-THUNDER BALL	A10
0929-COMIC BAKERY	A10
0931-BOUNDER	A10
0932-KNIGHT LORE	A10
0933-NIGHT SHADE	A10
0934-JACK THE NIPPER	A10
0935-TIME PILOT	A10
0937-VALKYR	A10
0938-ALIEN 8	A10
0939-GUN FRIGHT	A10
0940-HARVEY SMITH SHOW	
JUMPER	A10
0941-KING'S VALLEY	A10
0942-MAGICAL TREE	A10
0944-MOONSWEEPER	A10
0945-FORMATION Z	A10
0946-RALLY X	A10
0947-MOON PATROL	A10
0948-ZAXXON II	A10
0949-ELEVATOR ACTION	A10
0950-BASEBALL	A10
0951-YIE AR KUNG FU II	A10
0952-WARROID	A10
0953-LAS TRES LUCES DE	
GLAURUNG	A10
0954-T.Z.R.	A10
0955-SCARLET 7	A10
0957-PINE APPLIN	A10
0958-GANG MAN	A10
0960-ZANAC	A10
0961-EXOIDE Z	A10

096-RAID ON BUNGELIN BAY	A10
0963-SWEET ACORN	A10
0964-XIXOLOG	A10
0965-CIRCUS CHARLIE	A10
0966-HYPER RALLY	A10
0967-BOSCONIAN	A10
0968-KNIGHTMARE	A10
0969-Mr. DO	A10
0970-PINKY CHASE	A10
0971-SCION	A10
0972-CONGO BONGO	A10
0973-SAMURAI NINJA II	A10
0974-LDE RUNNER I	A10
0975-BRUCE LEE	A10
0976-GYRODINE	A10
0977-THE WAY OF THETIER	A28
0978-THE GOONIES	A10
0979-SKYGALDO	A10
0980-LAZY JONES	A10
0981-BLACK JACK	A10
0982-CRUZADER	A10
098-DUNKSHOT (BASKETBALL)	A10
0984-B.C. QUEST II	A10
0985-COASTE RACE	A10
0986-THEXDER	A10
0987-EXXA INNOVA	A10
0988-PROTECTOR	A10
0989-BANK PANIC	A10
0990-ZEXAS II	A10
0991-GUARDIC	A10
0992-HANG ON	A10
0994-ALCAZAR	A10
0995-LODE RUNNER II	A10
0996-FAIRY	A10
0997-HOLE IN ONE	
PROFESSIONAL	A10
0998-RABBIAN	A10
0999-RAMBO	A10

1000-STAR FORCE	A12
1001-MAGICAL KID WIZZARD	A12
1002-PAIRS	A10
1003-ARKANOID	A12
1004-GREEN BERET	A12
1005-FIRE HAWK	A10
1006-MOLE MOLE II	A10
1007-BATMAN	A12
1008-HEAD OVER HEELS	A12
1009-CABBAGE PATCH KIDS	A12
1010-DAMAS	A10
1011-SURVIVOR	A12
1012-GAVULET	A12
013-DONKEY KONG	A12
1014-PHANTOMAS 2	A12
1015-INTERNATIONAL KARATE	A12
1016-KNOCK OUT 3D	A12
1017-SPACE SHUTTLE	A12
1018-DEMONIA (GHOST'N'	
GOBLIUS)	A12
019-MASTER OF THE LAMPS	A12
1020-KONAMIS BOXING	A10
1021-007-THE LIVING DAY	
LIGHTS	A12

UTILITARIOS

0829-DESENSAMBLADOR	A30
0830-ENSAMBLADOR	A30
0831-CONTABILIDAD GENERAL	A10
0832-FICEROS	A10
0833-PASCAL	A30
0834-MINILOGO	A15
0835-PLANILLA DE CALCULO	A10
0887-COTROL DE STOCK	A10
0911-CONTROL BANCARIO	A10
0936-TASWORD II (Pr. de textos)	A15
0993-EDD I (graficador)	A42
1022-KIT DE ALINEACION DE	
GRABADOR	A30

TITULOS: _____

NOMBRE Y APELLIDO: _____

DIRECCION: _____

C.P. _____

LOCALIDAD: _____

FORMA DE PAGO: Cheque/giro

Contrarrembolso

PROVINCIA _____

Cheques y giros a la orden de Edmundo A. Goldin. Gastos de Envío A8.-

800-1300: Gráfico (inicio de cada juego).
870-970: Gráfico (total).

1350-1380: Subrutina sonido.
1420-1430: Subrutina impresión.

```

10 ON ERROR GOTO 1310
20 COLOR,1,1:SCREEN2:DIM BFICHA(11,4)
30 CIRCLE (125,30),30,6,0,2,1
40 CIRCLE (117,15),13,6,2,3,1
50 PSET(153,33),6
60 DRAW "NE2D20G7D25L20H10L10U5E3U4H2U2R5H5U1E2
  U2H3U1E6U4H4U3H3U15"
70 PAINT(125,30),6
80 FSET (25,145),10
90 DRAW "R20E5U20H5L15U10R20U5H5L20G5D20F5R15D10
  20D5F5BR35R20E5U5L20U30R20U5H5L20G5
  D40F5BR35R5U20R5F9D11R7U14H5E5U20H5L20G5D40F
  5BU30BR5R10U10L10D10"
100 PSET (133,145),10
110 DRAW "R5U20R10D20R5E5U40H5L20G5D40F5BR5BU30
  R10U10L10D10BR30BD30R20E5U20H5L15U10R
  20U5H5L20G5D20F5R15D10L20D5F5BR35R5U20R10D2
  0R5E5U40H5L5D20L10U20L5G5D40F5"
120 FOR R=1 TO 1000:NEXT R
130 SCREEN2:COLOR15,4,1
140 GOSUB 990
150 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
160 REM ELIGE LOS 4 COLORES
170 REM SECRETOS
180 REM
190 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
200 FOR S=1 TO 4:RESTORE 240
210 CS=INT(RND(-TIME)*5)+1
220 FOR CA= 1 TO CS:READ COL(S)
230 NEXT CA:NEXTS
240 DATA 1
250 DATA 6
260 DATA 10
270 DATA 12
280 DATA 15
290 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
300 REM PARTE 2
310 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
320 FILA=10:GILA=0
330 FILA=FILA+15:XCOL=10:GILA=GILA+1
340 FOR X=1 TO 4:XCOL=XCOL+20
350 ON KEY GOSUB 380,390,400,410,420
360 KEY(1)ON:KEY(2)ON:KEY(3)ON:KEY(4)ON:KEY(5)ON
370 GOTO 370
380 KOL=15:RETURN 430
390 KOL=12:RETURN 430
400 KOL=6:RETURN 430
410 KOL=10:RETURN 430
420 KOL=1:RETURN 430
430 CIRCLE(XCOL,FILA),5,KOL:PAINT(XCOL,FILA),KOL
440 BFICHA(GILA,X)=KOL:V=111:GOSUB1350
450 NEXT X
460 ZCOL=120:FOR R=1 TO 4
470 R1(R)=0:R3(R)=0:NEXT:R2=0:R4=0
480 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
490 REM
500 REM PARTE 3
510 REM
520 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
530 REM SELECCIONA FICHAS NEGRAS
540 REM PARA CONTROL
550 REM
560 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
570 CTRL=0
580 FOR R=1 TO 4
590 IF COL(R)=BFICHA(GILA,R)THEN 600 ELSE 620
600 ZCOL=ZCOL+10:CIRCLE(ZCOL,FILA),3,1:CTRL=
  CTRL+1:V=63:GOSUB1350
610 R1(R)=1:R3(R)=1
620 NEXT R
630 IF CTRL=4 THEN 640 ELSE 700
640 GOSUB1420
650 GOSUB 890
660 GOSUB 990
670 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
680 REM SELECCIONA BLANCAS
690 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
700 FOR R=1 TO 4

```

```

710 FOR R0=1 TO 4
720 IF R1(R)=1 THEN 770 ELSE 730
730 IF R3(R0)=1 THEN 770 ELSE 740
740 IF COL(R)=BFICHA(GILA,R0) THEN 750 ELSE 770
750 ZCOL=ZCOL+10:CIRCLE(ZCOL,FILA),3,15:V=63:
  GOSUB1350
760 R1(R)=1:R3(R0)=1
770 NEXT R0,R
780 GOSUB 1420
790 IF GILA=>11 THEN GOSUB 890 ELSE 330
800 GOSUB 990
810 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
820 REM
830 REM SUBROUTINA GRAFICO
840 REM Y MUESTRA DE CODIGO
850 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
860 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
870 LINE(103,0)-(119,192),1,BF
880 LINE(171,0)-(255,192),1,BF
890 LINE(21,0)-(101,19),4,BF:KL=10
900 FOR R=1 TO 4:KL=KL+20:CIRCLE(KL,9),5,COL(R)
  :PAINT(KL,9),COL(R):NEXT R
910 OPEN"GRP:"FOR OUTPUT AS#1:COLOR1,1,1
920 PSET(178,180):PRINT#1,"PULSO"
930 CLOSE:COLOR15,1,1
940 OPEN"GRP:"FOR OUTPUT AS #1
950 PSET(178,180):PRINT#1,"PULSE 1"
960 CLOSE
970 IF INKEY$="1" THEN 980 ELSE 910
980 RETURN
990 LINE(20,0)-(102,192),5,BF
1000 LINE(32,0)-(87,19),1,BF
1010 OPEN "GRP:" FOR OUTPUT AS #1
1020 PSET(40,1):COLOR15,4,1
1030 PRINT #1,"CODIGO"
1040 PSET(35,11)
1050 PRINT #1,"SECRETO"
1060 LINE(20,0)-(102,192),1,B
1070 LINE(0,0)-(19,192),1,BF
1080 FOR W=25 TO 175 STEP 15
1090 FOR X=30 TO 90 STEP 20
1100 CIRCLE(X,W),5,15
1110 NEXT X,W
1120 LINE(120,0)-(170,192),14,BF
1130 PSET(130,1):PRINT#1,"CTRL"
1140 PSET(178,150):PRINT#1,"NIVEL":CLOSE
1150 FOR W=25 TO 175 STEP 15
1160 LINE(123,0)-(166,9),15,B
1170 DY=20:OPEN "GRP:"FOR OUTPUT AS #1
1180 RESTORE 1240
1190 FOR H=1 TO 5:DY=DY+20:READ HD
1200 PSET(180,DY):PRINT #1,"F";H
1210 CIRCLE(220,DY+3),5,HD:IF H=5 THEN 1230
  ELSE 1220
1220 PAINT(220,DY+3),HD
1230 NEXT H
1240 DATA 15,12,6,10,15
1250 COLOR 1,1,1
1260 PSET(178,180):PRINT#1,"PULSO"
1270 PSET(220,150):PRINT #1,"F"
1280 CLOSE
1290 COLOR 15,4,1
1300 RETURN 200
1310 RUN
1320 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
1330 REM SUBROUTINA SONIDO @
1340 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
1350 SOUND1,0:SOUND0,V:SOUND7,&B111111110
1360 SOUND8,16:SOUND11,0
1370 SOUND12,1:SOUND13,9
1380 RETURN
1390 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
1400 REM SUBR. IMPR. NIVEL
1410 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
1420 OPEN"GRP:"FOR OUTPUT AS #1:COLOR1,1,1:PSET
  (220,150)
1430 PRINT#1,"NIVEL":COLOR14,4,1:PSET(220,150):
  PRINT#1,GILA:CLOSE:RETURN

```


3^{er} CONCURSO

DE PROGRAMAS

auspiciado por TELEMATICA S.A. que proveerá los siguientes Premios:

PRIMER PREMIO

UN PERIFERICO

(a elección entre un monitor, una diskettera y una impresora).



UNA BECA

para trabajar en el Departamento de Investigación y Desarrollo de Telemática S.A.

SEGUNDO PREMIO

UN PERIFERICO

(a elección entre un monitor, una diskettera y una impresora).



ESPECIAL

Entre los programas recibidos, algunos de ellos podrán ser editados por SYSTEMAC S.A., reconociéndose los derechos de autor

En caso de que el ganador no pueda utilizar la beca, será ofrecida a quien obtenga el segundo premio, y si éste tampoco pudiera aprovecharla se otorgará a alguno de los participantes del certamen que se hubiera destacado.

Se premiará el mejor software de cualquier clase (juegos, utilitarios, científico o comercial).

B A S E S: No sólo será indispensable que el programa enviado en caset ó disket funcione correctamente, sino que además debe cumplir con ciertas reglas:

- Programación estructurada en bloques fácilmente diferenciables.
- Fácil seguimiento del mismo y detalle de éste como parte de su documentación. (Diagrama de bloques con los números de línea que los identifiquen)
- Aclaración y clara explicación de los algoritmos utilizados, deben figurar como parte de la documentación.
- Las variables y/o direcciones de memoria utilizados también se deben incluir en esta documentación.
- Listado de nemónicos assembler y la localización en memoria si es que se utiliza este tipo de lenguaje.
- Calidad y originalidad de gráficos, sonidos y pantallas de menú.

Los trabajos deberán enviarse antes del 30 de noviembre próximo (cierre del certámen) a: Paraná 720, piso 5º, (1017) Capital Federal.

¿C EL LENGUAJE DEL FUTURO?

El lenguaje C está adquiriendo un auge muy grande en los círculos computacionales. Su velocidad y flexibilidad lo hacen ideal para la construcción de programas comerciales, sistemas operativos y hasta de programas hogareños.

Erase una vez... La historia comienza, como muchas otras que tienen que ver con el desarrollo de la computación, en los laboratorios Bell de la A T & T. Allí, ante el requerimiento de un sistema operativo poderoso para máquinas chicas, con capacidades similares a las de los grandes operativos de los Mainframes, Ken Thomson empieza a programar sobre un ensamblador hasta llegar a una primera versión de UNIX. Pero las necesidades son mayores aún. La tarea se complica tanto que Thomson decide utilizar un lenguaje de alto nivel (el nivel de un lenguaje tiene que ver con su grado de abstracción, es decir, con cuán fácil resulta hacer un programa; por ejemplo el más bajo nivel es el código máquina mientras que de alto nivel son BASIC o Pascal y uno de los más altos es el NATURAL de IBM, un sistema inteligente de Bases de Datos).

A la hora de elegir, prefiere desarrollar un lenguaje propio (cosa bastante común para esta gente), y crea una versión de un compilado de BCPL, un poco difundido hijo del Pascal de Niklaus Wirth. Abreviando lo bautiza B.

Pero hace falta más, y Thomson está en una encrucijada: trabajar sobre un ensamblador significa largas horas de codificación y se hace inmensamente tedioso y complicado; por otro lado, el B le facilita la programación pero baja considerablemente el rendimiento en velocidad y, sobre todo, genera objetos muy grandes.

Extenuado sobre la DEC-PDP11 (máquina clave también en la historia de la programación moderna), Ken pasa el problema a alguien de confianza: Dennis Ritchie.

Este señor tiene eso que los programadores suelen llamar "toque maestro", y lo que hace es empezar de nuevo a codificar en B. Pero mientras desarrolla el UNIX va modificando paralelamente el B hasta convertirlo en otro lenguaje al que llama, obviamente, C. Y aquí es donde Ritchie cambia la historia, prosiguiendo las enseñanzas y líneas de razonamiento de los grandes maestros: Dijkstra, Knutt y Wirth,



padres de la programación.

Parte de las rutinas mínimas en ensamblador hasta llegar a un protocompilador de C. A partir de ahí, el verdadero compilador, ¡está hecho en C! Mentas recursivas, si las hay...

EL COMPILADOR DE C

Una vez finalizado el Sistema Operativo UNIX, el C quedó definitivamente vinculado a éste. De manera que el UNIX es el entorno ideal para desarrollar software en C y C es el lenguaje ideal para cualquier implementación que deba ejecutarse sobre este S.O. Con el correr del tiempo, la potencia del lenguaje y sus singulares características le han hecho ganar un lugar privilegiado ante los ojos de los programadores.

Aunque es un lenguaje más de propósitos generales, el C ganó aceptación rápidamente a partir de su independencia del hardware. ¿Qué quiero decir? Que el C trabaja sobre objetos

sencillos y comunes a la mayoría de las computadoras. Originalmente, sus tipos de datos son solamente caracteres (de un byte), números y direcciones de memoria, y sus operaciones aritmético-lógicas son las mínimas existentes en cualquier máquina actual. No tiene estructuras como strings, sets o listas. Tampoco tiene implementadas funciones de Entrada/salida.

Eso hace que un compilador de C pueda ser implementado sobre cualquier computador. ¿Entonces, dónde está su potencia? En sus estructuras de control: bucles y condicionales (FOR, WHILE, IF) son en el C tan funcionales que la programación se hace sumamente dinámica; y por supuesto, las aparentes limitaciones tienen su porqué. Todas las funciones que no se hallan implementadas en el compilador mismo están incorporadas en las llamadas bibliotecas del C.

De esta manera se pueden utilizar todos los tipos de estructuras de datos

o funciones de entrada y salida que antes mencionábamos mediante llamadas a dichas bibliotecas. La gracia es precisamente que son llamadas solamente las necesarias para cada programa, lo que hace que el programa objeto no contenga rutinas que no se utilizan. Esto obviamente optimiza mucho el largo del código.

El C es, por ser tan conciso, bastante fácil de aprender, y algunos de sus recursos específicos hacen la delicia de los aficionados a la programación. Un ejemplo concreto es que cualquier rutina desarrollada por el usuario puede ser definida como una función más de la biblioteca, siguiendo así la tendencia moderna de los lenguajes extensibles. Y cuando el problema se pone difícil, o los tiempos de ejecución deben ser reducidos, existe la posibilidad de definir una biblioteca con rutinas en código máquina.

Si bien el C está hecho para correr originariamente bajo UNIX, esto no invalida su utilización en otros sistemas. Una de sus características principales es precisamente la transportabilidad. Utilizando las bibliotecas estándar el mismo programa fuente puede ser compilado y ejecutado en distintas computadoras. Y aun cuando fuera necesario incluir partes en código máquina, éstas, definidas como bibliotecas en el compilador original, pueden ser reescritas en el compilador de la máquina destino, trabajo que no puede compararse a tener que reescribir todo el programa.

APLICACIONES DEL C

La velocidad que alcanza y lo compacto que resulta el código generado hizo que muchas de las aplicaciones cuya resolución normalmente se plantearía en Assembler fueran resueltas utilizando C.

Es muy normal que el programador genere algunas bibliotecas en Assembler

para controlar el hardware sobre el que va a correr el programa, pero una vez definidas, éstas quedan incorporadas al compilador. Generando las mismas funciones de biblioteca para distintos entornos del programa que las utiliza puede ser recompilado en todos ellos, aumentando muchísimo la cantidad de usuarios potenciales.

Mediante estas bibliotecas uno va personalizando su compilador. Por ejemplo, podríamos crear una librería para gráficos (para poner un punto, u-



na línea, un círculo, definiríamos funciones: PUNTO (X,Y), LINEA (X1, Y1, X2, Y2), CIRCULO (X, Y, R) u otras o podríamos hacer una biblioteca de funciones para manejar archivos con estructura en árbol. De allí en más, cada vez que en un programa necesite hacer una operación de este tipo sólo debo avisar en qué fuente debe ser incluida tal o cual biblioteca, y luego utilizo mis funciones como cualquiera de las originales. Este método de trabajo es realmente cómodo y eficiente, lo que se comprueba al saber que gran parte del software famoso está escrito en C. Por ejemplo, el MS-DOS desde su versión 3.0 en adelante, el DBase III (y el III Plus), el LOTUS 1-2-3, y muchísimos más. Incluso la mayoría de los compiladores de otros lengua-

jes (Pascal, BASIC, Prolog, LISP, ensambladores), y por supuesto, los de C, están escritos en C.

EL MSX-C

Para adentrarnos un poco en el lenguaje mismo vamos a hablar del compilador de C que más nos interesa: el MSX. Este es un compilador igual a cualquier otro, con una biblioteca estándar, salvo por una limitación: entre las variables numéricas no existen los tipos FLOAT (punto flotante), ni los LONG INTEGER (enteros grandes, normalmente de 4 bytes). El único tipo numérico es el INTEGER, de 2 bytes de largo (puede contener números entre -32768 y 32767 o, declarándolo como UNSIGNED INTEGER, entre 0 y 65535). El núcleo del compilado contiene:

- Los operadores aritméticos "+", "-", "*", "/" y "%" (resto de la división), que admiten todo tipo de combinaciones, agrupados con paréntesis.
- Los relacionales ">", "<", "=", ">=", "<=", "==" (igual que), "!=" (distinto de).
- Los de manejo de bits "&" (and), "!" (or), "^" (xor), "<<" (rotación a la izquierda) y ">>" (rotación a la derecha).
- Y los maravillosos operadores de incremento "++", y decremento "--".

Estos operadores pueden usarse sobre una variable dentro de cualquier expresión; por ejemplo: "A=B++" carga en A el contenido de B y luego incrementa B en 1. Pero "A=++B" incrementa primero B y luego carga el contenido en A. Todo depende de la posición en que coloque el doble signo. Exactamente lo mismo puede hacerse con "--", que decrementa en 1 el contenido de la variable.

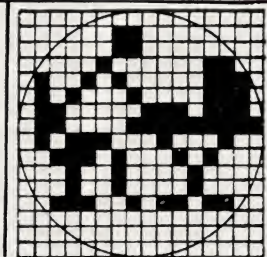
Las estructuras de control son similares a las de Pascal: WHILE, IF, FOR, DO WHILE (que, como el REPEAT del Pascal, evalúa la condición al final del ciclo), y existe una sentencia SWITCH que trabaja en forma similar al CASE. Las instrucciones que deben ejecu-

DELTA * tron taller de computación

Director:
Gustavo O. Delfino

651-4027

**CURSOS DE COMPUTACION
para adultos docentes
adolescentes y niños
BASIC-LOGO-UTILITARIOS**



CURSOS DE:
Introducción a la
Informática
Programación BASIC
Planillas de Cálculo
Procesador de Textos
Bases de Datos
Talleres LOGO para
niños y docentes
Servicio Integral de
Educación Informática
a Escuelas Primarias
Y Secundarias

CERVEUX

tarse según la condición evaluada en el bucle van encerradas entre llaves, aun cuando no estén en la misma línea, y cada una de ellas debe ser finalizada por ";" como en Pascal.

Cada una de las rutinas es definida como una función. Se llaman así porque pueden emitir un resultado y se le pueden pasar parámetros. Por ejemplo:

```
int potencia ( A , B )
int A , B;
{
  int P , Q;
  P = A;
  for ( Q = 0 ; Q != B ; Q++
    P * = A;
  return ( P );
```

Esta función eleva a la potencia B un número A. Al definirla, "int" especifica el tipo INTEGER para el resultado de la función. Al empezar, primero declaramos el tipo de los parámetros (en este caso también INTEGER), abrimos llaves y declaramos las demás variables que vamos a utilizar. Cargamos en P el valor de A y comenzamos el bucle FOR, donde, entre paréntesis y separadas por ";", van tres expresiones.

La primera es la inicialización del lazo; en este caso ponemos Q en 0.

La segunda es la condición que es evaluada antes de reiniciar un ciclo. Aquí preguntamos si Q es distinto a B.

Y la tercera expresión indica lo que debe hacerse al iniciar cada ciclo. En el ejemplo, incrementar Q.

Dentro del lazo vamos acumulando en P su propio contenido por A. La sintaxis de esta asignación es característica del C y es lo mismo que "P = P * A", que también sería válido. La idea es entonces que multiplicamos A por A, B veces. En este caso, como hay una sola instrucción dentro del bucle, no necesitamos encerrarla entre llaves. Una vez que salimos del bucle, ya podemos salir de la función, devolviendo el resultado que nos quedó en P; de eso se encarga la instrucción return(), donde entre paréntesis indicamos la variable que pasamos como salida de la función.

Si quisiéramos desde otra función guardar en una variable cualquiera (por ej: N), 14 elevado a la quinta, sólo tenemos que hacer:

N = potencia (14 , 5);

Las variables declaradas en una función pertenecen solamente a ésta y son independientes de las declaradas al principio del programa que son comunes a todas.

Una función también puede llamarse a

si misma, ya que el C admite recursividad. Pero como esto genera mucho espacio de reserva para los stacks de variables, podemos avisarle al compilador si no vamos a utilizar recursividad, y esos espacios no son generados. Este tipo de advertencias al compilador se indican con la sentencia PRAGMA: PRAGMA NONREC indica si no hay recursividad, SPACE OPT da prioridad al ahorro de espacio, aunque el código generado sea más lento, y SPEED OPT da prioridad a la velocidad de ejecución, aunque el objeto quede más largo.

LA BIBLIOTECA ESTÁNDAR Y LA COMPILACION

En esta biblioteca provista con el MSX-C, tenemos todas las instrucciones de Entrada y Salida, manejo de



strings, etcétera, que no contiene el compilador en sí mismo. La llamamos estándar porque contiene lo mismo que cualquier otro compilador de cualquier otra máquina, lo que hace posible la tan mencionada transportabilidad.

Las bibliotecas a incluirse son determinadas en el programa mediante la sentencia #INCLUDE, y una vez que el fuente está listo, se graba y se compila ejecutando desde el DOS el comando "C", que llama al "C.COM", el compilador.

Este compilador (la versión 2.0) genera un programa fuente en Assembler de 8080, un microprocesador anterior al Z-80, pero con el mismo código de máquina excepto por algunas instrucciones "extras" del Z-80. De todos modos tenemos noticias de que pronto estará la versión 3.0 que genera código Assembler de Z-80.

Lo cierto es que este programa Assembler debe ser ensamblado con el

Macro-Assembler 80 (M80) y "linkeado" con el Linker 80 (L80), para obtener el programa objeto que tendrá el mismo nombre del fuente pero con extensión ".COM", lo que permite ejecutarlo desde el Sistema Operativo.

Y ESTA HISTORIA CONTINUARA...

¿Qué puede pasar con el C?. Indudablemente este lenguaje tiene características muy interesantes que facilitan la labor del programador acostumbrado a trabajar con lenguajes estructurados. Además, su velocidad, levemente menor al Assembler, y su transportabilidad lo están llevando a desplazar al Assembler en la elaboración de sistemas operativos, compiladores y otras aplicaciones. La razón es muy sencilla: el tiempo de investigación que lleva el descubrir los secretos de un microprocesador no se justifica si en poco tiempo se debe comenzar el estudio de uno nuevo. Con el C, en cambio, los ajustes que hay que hacer son mínimos, justamente por la mencionada transportabilidad. En cuanto al usuario común, el C no es difícil de aprender y representa una buena opción para el que siente que el BASIC le está "quedando chico" y no quiere introducirse en el mundo del Assembler.

INGRESO DE TEXTO

Con esta rutina tratamos de solucionar un problema bastante común: el largo de las strings que deben ser ingresadas en un programa.

Esto es casi imprescindible si usan archivos relativos, ya que el usuario sabrá que todo lo que logre ingresar queda grabado.

El uso de la rutina es muy sencillo y

```
10 CLS
20 LOCATE 2,10:PRINT"Nombre:
30 QX=10:QY=10:QL=20:GOSUB 10000
40 PRINT"*";QA$;"*"
50 END
99993 "-----
99994 " -- Ingreso Controlado --
99995 "-----
99996 "-----Martin Salias-----
99997 "-----
99998 "-----Club de Usuarios MSX-----
99999 "-----
10000 QA$="":QP=0
10010 LOCATE QX+QP,QY,1
10020 QJ#=INPUT$(1)
10030 IF ASC(QJ#)=13 THEN PRINT:
      RETURN
10040 IF ASC(QJ#)=8 AND QP>0 THEN
      QP=QP-1:QA$=LEFT$(QA$,QP):GOTO
10080
10060 IF QP=QL THEN BEEP:GOTO 10010
10070 QP=QP+1:QA$=QA$+QJ#
10080 LOCATE QX,QY:PRINTQA$;" "
10090 GOTO 10010
```


todas las variables empiezan con la letra "Q", para evitar conflictos con el programa donde la utilice.

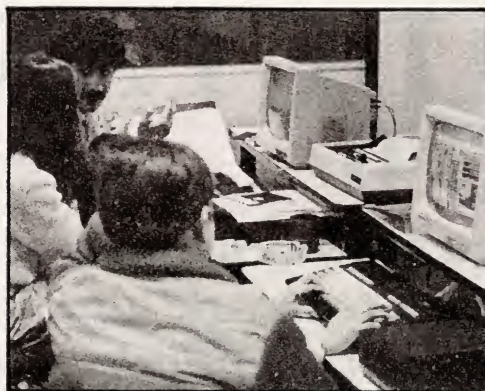
Tienen que pasar los parámetros QX y QY, que indican la posición a partir de la cual debe aceptarse el texto, y QL donde se especifica el largo máximo de éste. Al llamar a la rutina 10000, aparecerá el cursor como si fuera un INPUT común y corriente, salvo que al llegar al máximo hace un BEEP y no permite que continúe ingresando. Si algo me queda cortado, puedo volver hacia atrás con la tecla BS y corregirlo.

Les dejo como propuesta que expandan esta rutina agregándole, por ejemplo, el movimiento de los cursores, la validación del ingreso (si son números o letras) y un montón de cosas más que pueden ocurrírseles al usarla.

CURSO DE DIAGRAMACION ESTRUCTURADA

¿Qué es y para qué sirve?

Existen muchas formas de representar procesos lógicos; la diagramación es una de ellas. Parte de la base de



una simbología propia y tiene como premisa fundamental la representación de los pasos consecutivos y lógicamente ordenados necesarios para resolver un determinado problema. La utilización de esta simbología resulta práctica y en algunos casos necesaria (fundamentalmente en los principiantes) ya que simboliza procesos lógicos puros que no están perturbados por efectos secundarios, como lo son los de la sintaxis de los lenguajes de programación.

Resulta interesante aclarar que no consideramos necesaria a la Diagramación como paso previo a la programación, sino como un **método de apoyo** (muy importante) para mejorar el

desarrollo de procesos lógicos.

Dentro de la Diagramación existen diversas simbologías, pero todas están enmarcadas dentro de dos grandes grupos: lineales o estructuradas.

En nuestro caso adoptamos el segundo tipo debido a su mayor claridad y eficiencia en la representación de un problema.

Nuestro curso comienza profundizando estas ideas. Se explican el concepto de diagrama de lógica y las estructuras básicas de control: **secuencia, decisión e iteración**. También el concepto de variables y sus diferentes tipos, las operaciones aritméticas y lógicas, qué es una bandera, etcétera.

Luego comienzan a verse las estructuras de datos: archivos, tablas, acumuladores, contadores. Con estos elementos se comienza a diagramar utilizando decisiones y rutinas que se aplican a cortes de control, actualización de archivos, manejo de tablas y matrices, y manejo de pantalla.

Todo esto es **universal** en cualquier lenguaje de programación que elijamos para trabajar, y conociendo la forma de resolver problemas lógicos, podremos dominar mucho más fácilmente cualquiera de ellos.

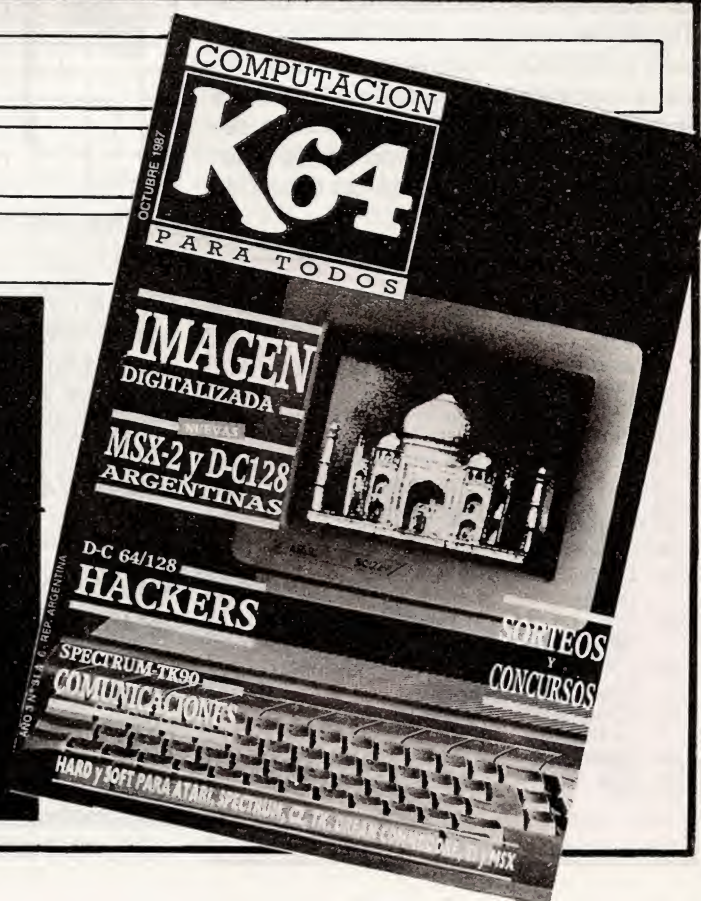
EN K-64 DE OCTUBRE

PROGRAMAS, TRUCOS Y

SUGERENCIAS PARA APROVECHAR

NUESTRA COMPUTADORA MSX.

- Con una computadora MSX2, una cámara y un digitalizador, se reproduce cualquier imagen
- Procesador de textos MSX-Write: las eñes y acentos.
- Inédito para MSX: "Krypton", un apasionante juego.
- Lápiz óptico.
- CONCURSOS y SORTEOS, con premios por valor de A 13.000



MANEJO DE PANTALLA EN MODO 0

Para muchos, la norma MSX es la natural heredera de la TI-99/4A. Explicamos cómo a través de aquella se han rescatado y mejorado la mayoría de sus cualidades como por ejemplo las facilidades para realizar gráficos desde BASIC en modo 2.

Es indudable que la gran mayoría de usuarios que hoy día transitan los senderos de la microcomputación, y que tienen más de 3 años de experiencia en el tema, han dado sus primeros pasos de la mano de la TEXAS TI 99/4A.

Quienes han seguido la evolución más lógica, han continuado el camino con la norma MSX, que vendría a ser su natural heredera. Dado que de la TI99 se ha tomado el chip de video (TMS 9919/29) el manejo de pantalla que poseen las microcomputadoras MSX es muy similar al de la Texas, y el modo 1 (SCREEN 1) que permite trabajar en 32 columnas y con SPRITES, es un fiel reflejo de sus características.

En las micro MSX se han rescatado la mayoría de las cualidades que la TI99 ofrecía sobre los demás equipos de su época, y así se hicieron muchas mejoras en ciertos casos con respecto al sistema original. Como ejemplos se pueden mencionar las facilidades para la realización de GRAFICOS desde BASIC en el modo 2 (SCREEN 2) de ALTA RESOLUCION, o la facilidad para cambiar de modos tanto desde los programas en BASIC, con el comando SCREEN ..., como desde el teclado mediante el uso de las teclas de

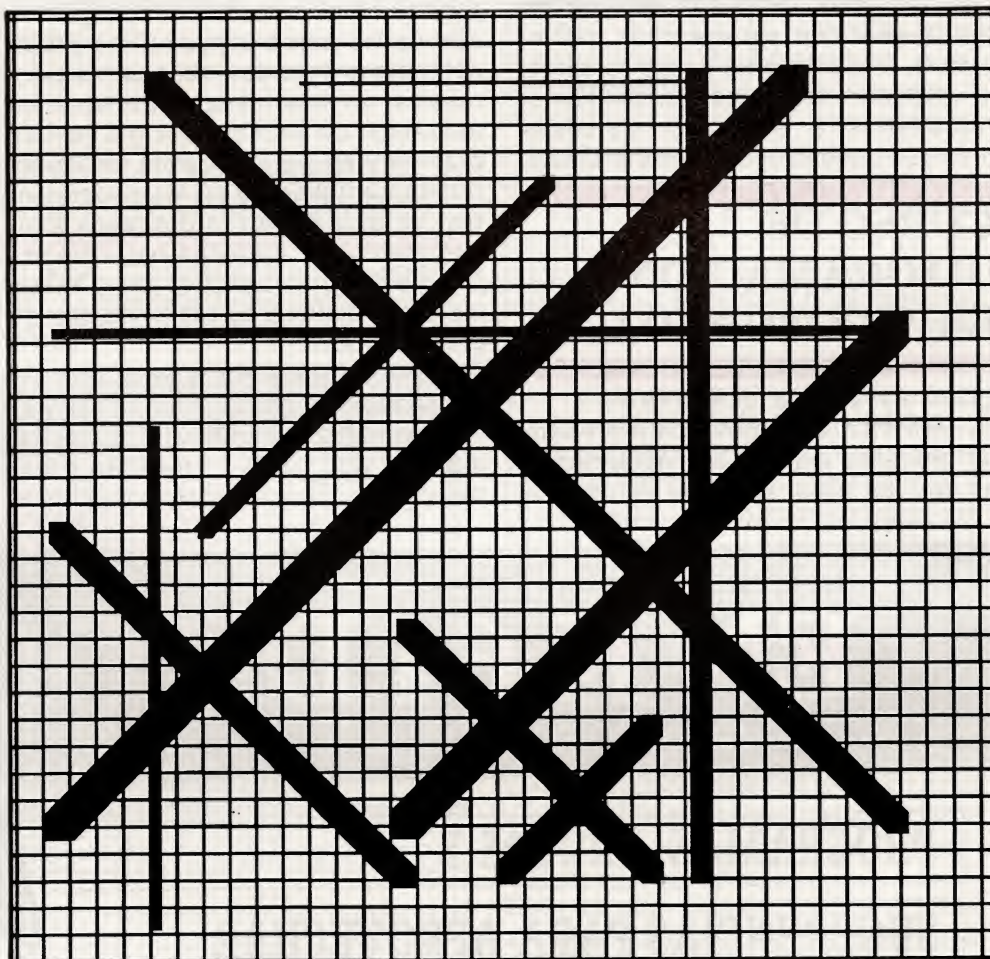


FIGURA 1

```

10 REM MANEJO PANTALLA SCREEN 0
12 REM -----
14 REM funcion call gchar TI99
16 REM
18 REM por Gustavo O. Delfino
20 REM
22 DEF FN GCHAR(C,F)=VPEEK(F*40+
  C+INT((40-PEEK(&HF3B0))/2+0.5))
24 SCREEN 0:WIDTH 37:COLOR 1,3
26 GOSUB 48
28 GOSUB 68
30 LOCATE 0,17:PRINT "FILA:
  Desde ";:INPUT FI
32 LOCATE 21,17:PRINT "Hasta ";:
  INPUT FF
34 LOCATE 0,18:PRINT "COLUMNAS:
  Desde ";:INPUT CI
36 LOCATE 21,18:PRINT "Hasta ";:
  INPUT CF
38 GOSUB 86
40 END
  
```

funcion que pueden ser reprogramadas (1).

Pero en todo cambio se pierden algunas cualidades, y más cuando dicho cambio se produce de la mano de otra empresa, en este caso la MICROSOFT Corp. quien adaptó su M-BASIC a las MSX, sin reparar demasiado en el excelente EXTENDED BASIC que traía la TI99

EL MANEJO DE PANTALLA EN LA TI99

En el camino quedaron pues ciertos comandos que posibilitaban un excelente manejo de pantalla. Por ejemplo el DISPLAY AT, que permitía no solo ubicar con precisión el cursor para escribir, sino además agregar un BEEP,

blanquear ciertas posiciones de la pantalla o limpiarla por completo; el ACCEPT AT, mucho mas versátil que el INPUT, permitía aceptar el ingreso de datos desde cualquier lugar de la pantalla, emitir un BEEP cuando la máquina estaba lista para el ingreso, blanquear ciertas posiciones de la pantalla o limpiarla por completo, limitar la cantidad de caracteres y el tipo de los mismos (numéricos, alfanuméricos, dígitos y especificar ciertos caracteres), y otros más de no tanta importancia.

Pero con un poco de imaginación, las funciones que mencionamos previa-

FIGURA 2

```

DEF FN GCHAR(C,F)=VPEEK(F*40+C+
  INT((40-PEEK(&HF3B0))/2+0.5))
  
```


PANTALLA EN MODO 0 - TEXT OS - SCREEN 0
40 COLUMNAS x 24 FILAS

VRAM

0
959
2048
4096
16383

A
0
1
2
3
39
40
920
959

FILA
0
01
23

VPEEK (2048 + 8 * ASCII + BYTE)

GRAFICO

TABLA B

BINARIO
00100000
01010000
10001000
10001000
11111000
10001000
10001000
00000000

DECIMAL
32
80
136
136
248
136
136
0

WIDTH 37

16 KBYTES * 1024 = 16384 BYTES DE VRAM

Las microcomputadoras que han adoptado la norma MSX en el mundo poseen una memoria RAM especialmente diseñada para almacenar los

FIGURA 4

```

44 REM SUBR. ENFILAR+ENCOLUMNAR
46 REM -----
48 FOR F=0 TO 23
50 VPOKE F*40, VAL (MID$(STR$(F),
  LEN$(STR$(F)), 1))+48
52 NEXT F
54 FOR C=0 TO 39
56 VPOKE 920+C, VAL (MID$(STR$(C),
  LEN$(STR$(C)), 1))+48
58 NEXT C
60 RETURN

```

datos relativos a la pantalla, tanto en lo referente a textos (caracteres alfanuméricos y símbolos especiales), como en colores, gráficos y sprites.

Esta memoria no es manejada directamente por el microprocesador Z80, sino por el chip de video TMS 9919 de Texas (VDP = Video Display Processor) liberando al Z80 de esta trabajosa tarea. La capacidad de almacenamiento es de 16 kbytes (16 x 1024 = 16384 bytes) y en el modo 0 de pantalla, que solo permite el empleo de textos, solamente ocupa alrededor de 3 kbytes, divididos en dos TABLAS.

TABLA DE CARACTERES: La primera de estas tablas guarda el registro de cada una de las 960 celdas de las que se puede disponer en este modo (40 columnas x 24 filas = 960 celdas). En ella se almacenan los códigos ASCII de los caracteres que la máquina debe exhibir en pantalla. La dirección de comienzo está dada por el valor de BASE (0), predefinida en MSX y que es 0, y se extiende hasta el byte 959. **TABLA DE PATRONES DE CARACTERES:** La segunda de las TABLAS del modo 0 es la correspondiente a los patrones de los 256 caracteres disponibles (ver Figura 3) cada uno de los cuales está definido por ocho bytes (del 0 al 7). La dirección de comienzo de esta tabla está dada por BASE(2) y es 2048, y se extiende hasta el byte 4096. Longitud 4096 - 2048 = 2048 bytes, correspondientes a 256 caracteres x 8 bytes c/u. Para poder escribir en esta memoria VRAM (VIDEO RAM), se utiliza el comando VPOKE cuya sintaxis es: VPOKE (dirección, código ASCII), y para leer empleamos el VPEEK, comando cuya sintaxis es: VPEEK (dirección).

Si por ejemplo queremos colocar una Z (código ASCII=90) en el casillero correspondiente a la fila tercera (0-1-2...) y a la columna cuarta (0-1-2-3...), ingresaremos:

VPOKE (2*40+3,90) ---> Z VPOKE (F*40+C,ASCII)

Por el contrario, si deseamos leer el código del carácter que ocupa un determinado lugar en la pantalla, por ejemplo fila tercera, columna cuarta, tipearemos: PRINT VPEEK(2*40+3+2)

FIGURA 5

```

64 REM SUBR. LLENADO PANTALLA
66 REM -----
68 FOR F=0 TO 15
70 FOR C=2 TO 38
72 VPOKE F*40+C, INT (RND(1)*26+65)
74 NEXT C
76 NEXT F
78 RETURN
80 REM -----

```

---> 90 PRINT VPEEK(F*40+C+N)

Destaquemos que la columna que establecemos por medio de VPOKE corresponde al ancho de 40 (WIDTH 40) que puede no corresponderse con el ancho de pantalla que hayamos fijado. En la definición de la función se ha previsto esto, y agregamos a F * 40 + C, un sumando (N) que agrega las columnas que puedan faltar en la pantalla: INT (40-PEEK(&HF3B))/2+0.5), donde PEEK(&HF3B0) nos devuelve el ancho establecido por WIDTH. Por ejemplo: si trabajamos con un ancho de 37 (el estándar cuando encendemos el equipo), quedan 2 columnas a la izquierda - no disponibles - y 1 columna a la derecha.

ENFILAR Y ENCOLUMNAR

Para poder visualizar más fácilmente los efectos de la función GCHAR, preparamos esta sencilla subrutina que denominamos "ENFILAR Y ENCOLUMNAR", cuya misión consiste en colocar en la columna 0 (no disponible para el usuario; trabaja con un "WIDTH 37") el segundo dígito de la FILA correspondiente; y en la fila 23, cuya dirección comienza en 920 (ver Figura 4), también el segundo dígito pero de la COLUMNAS.

Nótese cómo, con un poco de imaginación, podemos fácilmente expandir notablemente los recursos -escasos- que nos brinda el BASIC.

EL LLENADO DE LA PANTALLA

Al efecto de facilitar aún más la comprensión del uso y las posibilidades que brinda la función GCHAR, agregamos esta subrutina auxiliar de la figura 5, que llena varias filas (16) de la pantalla con letras al azar, lo cual sumado al encolumnamiento y al enfilar, nos permitirá ejercitarnos adecuadamente.

Obsérvese que en la línea 70 los límites de la pantalla disponible, cuando trabajamos con un ancho de 37 columnas (WIDTH 37), están dados por las columnas 2 (la tercera empezando desde el margen izquierdo) y por la 38

FIGURA 6

```

82 REM SUBR SALIDA IMPRESA
84 REM -----
86 FOR F=FI TO FF
88 FOR C=CI TO CF
90 LPRINT CHR$(FN GCHAR(C,F));
92 NEXT C
94 LPRINT CHR$(10);
96 NEXT F
98 RETURN

```

(la segunda desde el margen derecho).

LA SALIDA POR IMPRESORA

Esta subrutina cumple en brindarnos una copia impresa del texto existente en la "VENTANA" delimitada por el usuario. Estrictamente hablando no es este el concepto adecuado de ventana, sino que más bien deberíamos referirnos a "PANELES".

Para quienes aún no sean propietarios de una impresora, sugerimos no amedrentarse por ello y concurrir, munidos de un paquete de masas para la secretaría, al CENTRO DE ASISTENCIA AL USUARIO de TALENT MSX más cercano a su domicilio, llevar el programa grabado en casete o disquete, y probarlo.

DESAFIOS A LA IMAGINACION

Queda planteado un desafío para los lectores de **LOAD**, que consiste en perfeccionar la Subrutina de SALIDA IMPRESA, de modo tal que respete en el papel, la posición relativa del PANEL delimitado en la pantalla.

Otra propuesta es la de reprogramar alguna tecla de función para que, al pulsarla, se cargue (MERGE) una rutina como la que hemos presentado, obviamente sin la parte de llenado de pantalla, con lo cual se podría hacer un vuelco por impresora de todo o de parte de la pantalla.

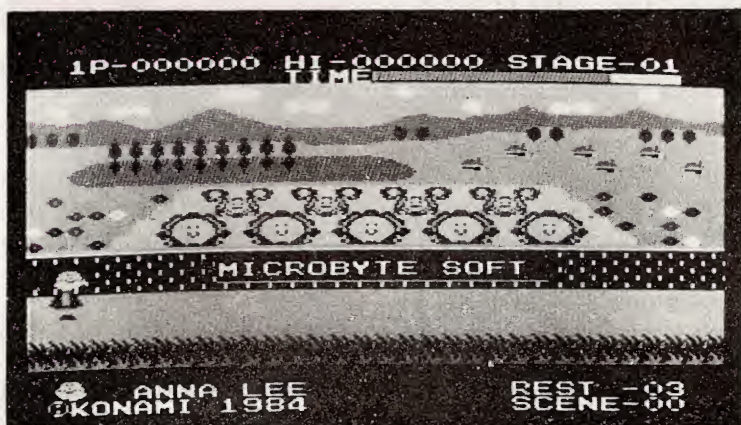
En este caso se deberían reubicar los mensajes de las líneas 30 a 36, para que no interfieran con la pantalla. Otra posibilidad sería la de reprogramar otra tecla de función para que nos permita almacenar en casete o disquete la pantalla de nuestro interés, siempre recurriendo a la función GCHAR que hemos presentado.

Esperamos recibir propuestas al respecto del Club de Fans de la MSX. Las iremos comentando en futuras colaboraciones.

(1) - Ver en **LOAD MSX** de Junio/1987 la Sección BUZON.

Gustavo O. Delfino

CABBAGE PATCH KIDS



lla de pozos, o unas arañas que quieren posarse sobre nuestra cabeza. Párrafo aparte merece la música. Pegadiza, melodías variadas y efectos sonoros

puntaje en forma de "bonus", por lo que debemos tratar de apurarnos.

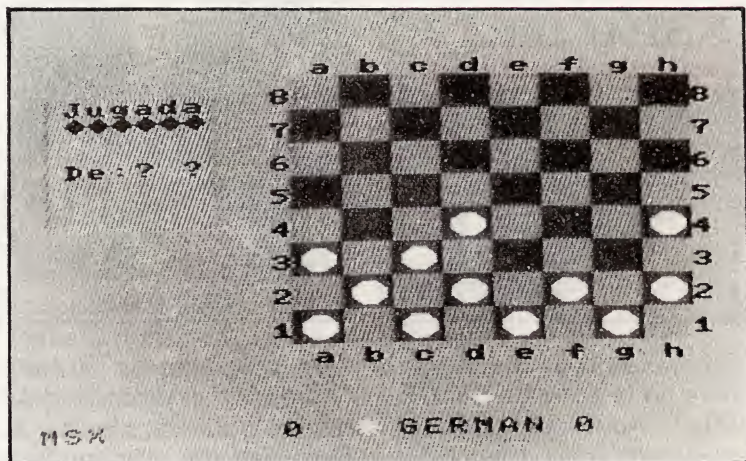
Otra dificultad es que hay un monstruito (a la manera de los Pac-man) que nos va cerrando el camino con la "non sancta" intención de comernos (devorarnos, deglutirnos, en otras palabras, que seamos su banquete).

Los diferentes niveles del juego alternan tres clases de dibujos (frutas, hortalí-

personas (al mismo tiempo), siendo esta última variante muy entretenida ya que fomenta un sano espíritu de competencia.

Sobre la música podemos decir que comenzamos odiándola pero cuando terminamos de jugar ya estábamos tararándola. (MICROBYTE)

DAMAS



Por un largo sendero una niña debe sortear obstáculos para continuar avanzando.

Al comienzo podemos elegir a nuestro personaje (cara, peinado, vestido y nombre). Todo al compás de una de las pegadizas melodías que tiene el programa.

Anna Lee, que es el nombre de la niña si no se lo modificamos, debe ir de pantalla en pantalla evitando ser arrollada por las piedras que vienen rodando y saltando por sobre un perro que está tendido en el medio del camino.

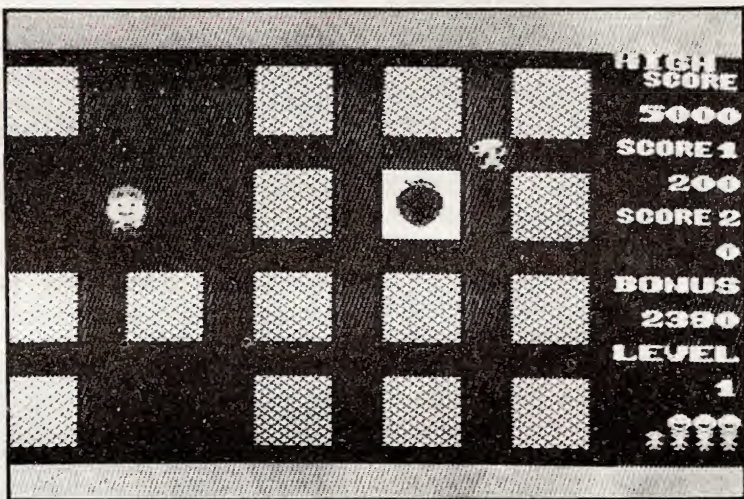
Hay también otro tipo de obstáculos que están situados en serie: troncos y pozos. Además hay que pasar un lago viajando en una isla flotante, a la que accedemos mediante una liana.

Otras pantallas tienen obstáculos combinados. Se llega a la última del primer nivel (9), que consiste en cuatro trampolines sobre los que hay que saltar para recoger una fruta.

Cumplido esto accedemos al siguiente nivel no sin antes hacer unos saltitos de festejo. Las complicaciones, como es de suponer, van en aumento. Aparece un pájaro con cara de malo, una fogata y fuentes que debemos saltar. También hay visitantes nuevos como unos peces saltarines que debemos evitar en la panta-

oportunos hacen intuir un trabajo muy serio en la realización de este programa. Es un juego muy "piola" para chicos de 10 a 12 años, pero se pueden "prender" los mayores si se resignan el pasar por las primeras pantallas que son muy fáciles. (MICROBYTE)

PAIRS



Un juego del tipo del "encuentre su pareja", en el que debemos ir dando vuelta los diferentes bloques (de a uno) y memorizando qué figuras se hallan debajo de ellos, para que, al descubrir dos iguales, y destaparlos en forma consecutiva, los mismos sean retirados del tablero y se nos otorguen puntos por ello. Para limpiar todo el panel hay un tiempo límite. El excedente de ese tiempo también se vuelca en nuestro

zas y animales) lo que permite a los chicos ir familiarizándose con estos conjun-

tos a la vez que ejercitan la memoria visual.

Los bloques a destapar son veinte: nueve parejas de objetos, un comodín, y un objeto que queda circunstancialmente libre y hace pareja con el comodín.

A medida que superamos niveles, los hambrientos monstruos se hacen más inteligentes (nos acorralan mejor), más veloces y, lo que es peor para nosotros, crecen en número.

Pueden jugar una o dos

Sentarse frente a la computadora y tratar de derrotarla es la propuesta de este programa.

El juego es las damas, más precisamente la versión del mismo que se juega sobre un tablero de ocho por ocho (como el del ajedrez).

Al comenzar nos pregunta nuestro nombre y si queremos jugar primero. Luego el juego se desarrolla normalmente, siendo bastante interesante el nivel de juego de la máquina, lo que le posibilita dar varios sustos, y hasta gana, si uno se descuida.

El programa tiene además una protección para que no podamos ingresar jugadas imposibles. Las reglas del juego son las tradicionales. Si no comemos una ficha enemiga que está en el aire, la máquina automáticamente "soplará" por no haberlo hecho y nos quitará nuestra ficha (¡mala!).

Cuando llegamos con una ficha propia a la octava fila, ésta se cambiará por una dama.

Pudo haberse mejorado un poco la presentación del juego colocando instruccio-

nes y diagramando un poco mejor la pantalla de presentación. Pero el programa cumple su objetivo, que es presentarnos lucha en una partida de damas, por lo que lo recomendamos a los que gustan de este juego. (MICROBYTE)

LA MOTO ESPACIAL 1



Este es uno de los programas de la serie de SYSTEMAC "Aprender jugando", destinada a los más chiquititos. Esta serie consiste en un conjunto de programas educativos dirigidos a niños de cuatro años en adelante. Asimismo algunos juegos pueden ser utilizados por niños de tres años, como también por aquellos que, superando los seis años presenten problemas de aprendizaje... La intención de esta serie es que los niños desarrollen su lenguaje y adquieran conocimientos al mismo tiempo que juegan con su computadora.

La Moto Espacial 1 pretende desarrollar la capacidad de observación en base a la formación de conjuntos.

Al comenzar el juego se propone una actividad. Luego, una vez resuelta, el niño recibe informaciones visuales y sonoras que le indican aciertos y errores. En el caso de que se equivoque, el niño tendrá otra oportunidad (o varias si es necesario), hasta que logre un acierto.

Así irá descubriendo que un conjunto es una colección de objetos que tienen una característica común. También desarrollará su destreza manual y su sentido espacial al tener que mover u-

na nave para posarla sobre el objeto elegido y transportarlo por una pantalla hasta el lugar adecuado. (SYSTEMAC)

LA ALFOMBRA MAGICA 1



Otro programa de la serie "Aprender jugando" en el que se pretende eliminar la pasividad del niño frente a la máquina y estimular la participación activa en el juego. Simultáneamente se le enseña y se lo adapta al manejo de la computadora.

"Aprender jugando" presenta una serie de actividades graduadas por dificultad cuyo objetivo es que el niño realice un aprendizaje progresivo, es decir, que utilice los conocimientos que posee como base del aprendizaje de otros nuevos.

La Alfombra Mágica 1 tiene como objetivo iniciar al niño en la expresión numérica por medio de juegos de identificación de las nueve primeras cifras.

Se busca: 1. Identificar la cantidad de objetos en los conjuntos dados. 2. Reconocer las cantidades que representan las grafías de los números. 3. Asociar conjuntos de objetos al número que lo representa. 4. Desarrollar la coordinación óculo-motriz.

El niño recibe información de sus aciertos o errores por medio de un código de color: verde si no comete errores; amarillo, si sólo comete uno, y rojo, si se equivoca más veces. Este código sirve además, como es

evidente, para que el niño vaya conociendo el código de señales de tránsito. (SYSTEMAC)

AVENTURAS EN EL CIRCO 1

También de la serie "Aprender jugando". Aventuras en el Circo 1 pretende desarrollar la capacidad de observación por medio de juegos de diferenciación de colores. El objetivo de este juego es que el niño forme parejas de objetos iguales en todas sus cualidades (color, forma, tamaño, posición). Al emparejar objetos se prepara para realizar actividades de clasificación posteriormente. Contribuye a ello, además, el dibujo



simbólico que aparece en pantalla para representar cada tema (conjunto o clase de objetos) por medio de distintos elementos relacionados entre sí por alguna característica común.

Los objetos se agrupan en diez temas diferentes: la familia, los transportes,

ce saltando sobre ellas. El juego tiene varios niveles, cada uno más difícil que el anterior. (SYSTEMAC)

GREEN BERET

Este juego de combate consiste en rescatar a soldados que han sido capturados por el maldito enemigo. Para cumplir esa misión debemos enfrentarnos con los agentes y las tropas que tienen órdenes de destruirnos.

Los soldados armados nos lanzan granadas y nos disparan a quemarropa. A ellos hay que esquivarlos, agachándose cuando vienen las balas y golpeándolos letalmente cuando los tenemos cerca. Luego están los agentes secretos. Unos, los oscuros, quieren prendernos por las buenas. A éstos es bastante fácil eliminarlos con uno o dos golpes. Pero luego aparecen los otros, blancos, que cuando nos ven saltan sobre nosotros cual tigres. La única forma de eliminarlos a estos agentes es evitar que nos ganen de mano en el salto.

El juego presenta mucha acción, en variados escenarios. Cada pantalla tiene un piso superior y uno inferior, y unas escaleritas que nos permiten pasar de uno a otro.

La estrategia adecuada es



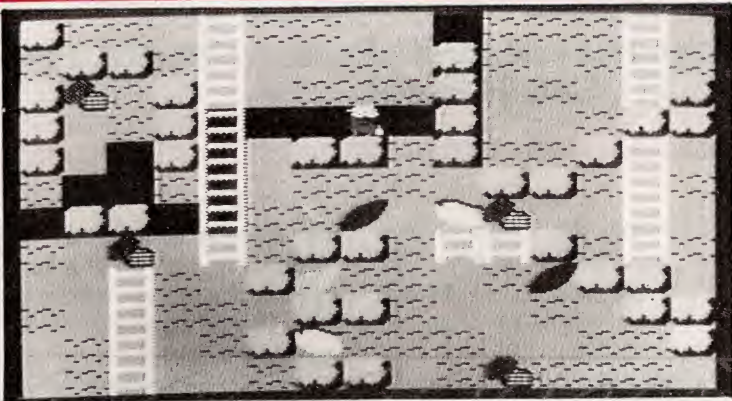
prendas de vestir, alimentos, flores, animales salvajes, animales domésticos, la casa, la escuela y la ciudad.

El payaso Plín Plín deberá trasladar los objetos de un lado a otro tratando de esquivar pulgas, cosa que ha-

combinar el tránsito hacia el objetivo alternando los dos pisos, y luchar sólo cuando sea necesario.

Los juegos de este tipo tienen una gran aceptación y éste, creemos, tiene las condiciones para "capturar" adeptos. (MICROBYTE)

MOLE MOLE 2



Un gordito narigón debe ir recogiendo los más variados objetos, la mayoría comestibles: bananas, cerezas, pedazos de tartas, diamantes, entre otros.

Su camino lo debe establecer atravesando fardos de pastos, que va deglutiendo. El juego es semejante al Boulderdash, pero aquí el camino del hombrecito no es tan libre. Por ejemplo, si quiere subir sólo puede hacerlo por las escaleras, por lo que el peligro de quedar atrapado en la parte inferior de la pantalla es muy grande.

Hay que pensar bien el recorrido que vamos a efectuar. Para eso tenemos un mapita al que accedemos apretando el botón del joystick.

Los niveles del juego aumentan en dificultad a medida que avanzamos, pero la buena noticia es que podemos elegir por cuál nivel empezar (no debemos "tragarnos" las mismas pantallas siempre).

(MICROBYTE)

ADMINISTRADOR DE VIDEO-CLUB

La proliferación del video en los últimos años es comparable en su campo al que tuvo la computación en el suyo.

No es raro entonces que muchos video clubes hayan decidido instalar ya una computadora que maneje la información de que disponen. Sin embargo en muchos casos la instalación de un sistema de este tipo es muy "casera" y los problemas comienzan a aparecer.

LOGICIEL ha creado este sistema administrador de video club que está diseñado para trabajar con un catálogo de hasta 1500 títulos de películas y un archivo de hasta 500 clientes.

Dispone de todas las posibilidades necesarias para el buen manejo de cualquier video club: altas, bajas,

consultas y modificaciones de películas y clientes; posibilidad de incorporación de nuevos clientes con o sin inscripción; listados alfabéticos y numéricos de clientes y títulos; listados de títulos por género; alquiler y devolución de títulos controlados por fecha; control listado de reservas; control de inscripciones; emisión de factura; listados de cumpleaños de los clientes, de deudores de películas y facturas, y de nuevas películas; depuración de archivos; posibilidad de manejo con hasta seis listas estándar de precios más una especial; control de ingresos y egresos, de cheques y de recargos por mora en la devolución; manejo de proveedores; cierre de caja del día; calculadora en todos los menús; posibilidad de aceptar un módulo de estadísticas.

El sistema consta de dos ti-

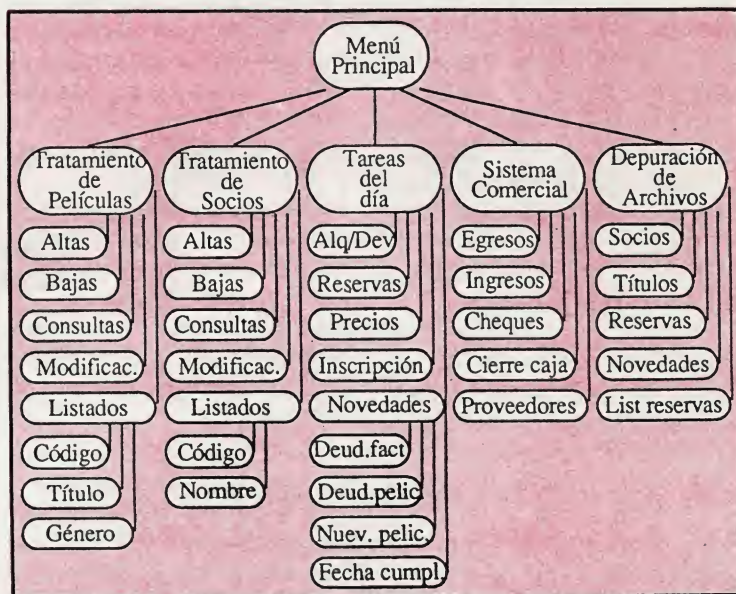
pos de pantallas: unas donde se selecciona la aplicación deseada (menús) y otras donde se ingresan datos para su procesamiento (pantallas de entradas de datos).

En el diagrama vemos al sistema en su conjunto y, de manera clara, la forma en que trabaja.

Restan hacer algunas aclaraciones. En el rubro "reservas" se ingresa desde qué día a qué otro se produce la misma (por ejemplo, desde el 10 hasta el 12). El sistema automáticamente volverá a habilitar la película para su alquiler si el día fijado (10) no fue retirada la misma.

Las seis listas de precios están pensadas para que pueda manejar precios distintos por 24, 48 y 72 horas.

(LOGICIEL)



MICRO SHOP

TECLADOS-DRIVES-MONITORES-IMPRESORAS
MANUALES de UTILITARIOS, COMERCIALES y JUEGOS
SOFT de BASE y APLICACION PARA PC, XT-AT

Envíos al Interior. Ventas por mayor y menor

Talcahuano 443 C. P. (1013) T. E. 35-6360

SOFT MSX

VIDEO CLUB
SISTEMA ADMINISTRADOR

CONTABILIDAD GENERAL
Conózcalo en Amatrix Bolívar 173
ó en Cúspide -Suipacha 1045 Cap.

DISTRIBUYE: ORGANIZACION MARO
C.C. N 21 (1826) Prov. Bs. As.
Tel: 242-5569 (zonas disponibles)

SINTETIZADOR DE VOZ

Recibimos varias cartas de lectores que han tenido problemas con este programa. Hemos trasladado las dudas al autor del mismo, y aquí está su respuesta: Debido al creciente rumor de las huestes enfurecidas que reclaman mi cabeza he decidido escribir estas líneas para salvar mi integridad física y calmar a los lectores, víctimas de la lluvia de DATA's de dicho programa.

Fernando Ramón y Luis López, de Necoché, son poseedores de una TOSHIBA HX-20, y cuando corren el programa se le resetea la máquina.

Es lógico que esto ocurra, ya que el programa está preparado para correr en Talent, cuyos 64 K están ubicados en el slot número 1 mientras que en la Toshiba dicha memoria ocupa otro slot (el 3). Al ejecutarse la rutina de la figura 5 (ver programa), este traslada el programa a la página 1 del slot 1. Como en la Toshiba este slot está vacío el programa se "esfuma".

Para solucionar el problema deberán hacer lo siguiente: reemplazar en la línea 600 del listado BASIC, contando desde la posición del primer DATA (03), el valor 54 por FC (posición 22), 50 por F0 (posición 28), y DA por FA (posición 34). Estos cambios habilitan el slot número 3.

Yendo a la inquietud de Leornado di Biase, de Capital, seguramente en su caso está faltando algún DATA,

lo que hace que la suma de control adicione algún valor incorrecto, produciendo una suma fuera de rango (OVERFLOW).

Con respecto a la misiva de Roberto Chushtek, de Tucumán, para su pesar le informo que el programa ha sido correctamente transcripo en la revista, y que el mismo, aunque parezca mentira,...FUNCIONA!!

Personalmente me he tomado el trabajo de controlar toda las DATA's, no encontrando ninguna diferencia con el listado original. Al efectuar dicha comprobación noté que puede confundirse fácilmente el número "8" con la letra "B". Tomen precauciones con este detalle. Casualmente Roberto tiene una diferencia 31469 a 31466 en la primera suma de la comprobación, que es justamente 3 (exactamente la diferencia entre 08 y 0BH (11)).

Para terminar la única salvedad que he de hacer es que falta una sentencia STOP en la línea 80 (80 IF S<>T THEN BEEP:PRINT** ERROR BLOQUE *:"M:STOP).

Este "BUG" no afecta el normal funcionamiento del programa. La sentencia STOP se necesita para detener al BASIC si se comprueba alguna diferencia en las sumas de verificación.

Espero haber disipado las dudas de muchos y que todos puedan hacer hablar a la "MSX".

EDUARDO LUIS
BLOTTA

CORREO INTERNACIONAL

Recibimos desde España una carta de un grupo de jóvenes inte-

Para comunicarse con nosotros deben escribirnos a "Load Revista para Usuarios de MSX", Paraná 720, 5to. Piso, (1017), Cap. Federal.

resados en máquinas de la norma MSX. Aquellos que quieran tener contacto con ellos e intercambiar programas deben escribir a:

GABRIEL MEDINA
VILCHEZ
c/PERU ,6, 1º
18600 MOTRIL
(GRANADA) ESPAÑA

CONCURSO

Me gusta mucho su revista y sobre todo los concursos. Mi pregunta es sobre el tercer concurso de programas. Me gustaría saber que factores se tienen en cuenta para determinar a los ganadores.

PABLO C. SBARRA
Tandil

Load MSX

Los ganadores se determinan luego de un análisis integral del programa. Previamente se hace una selección en base a calificación de distintos ítems como técnica de programación, documentación, utilidad, originalidad, efectos gráficos y sonoros entre otros. Para mayor información te recomendamos que leas las bases del concurso en donde se da una serie de requisitos que deben cumplir los programas.

BALANCE FAMILIAR

Soy lector de la revista desde hace algunos números y los felicito por el nivel que tiene.

Mi pregunta ¿cuál es el orden de copiado y ejecución del programa "Balance Familiar" que apareció en el número de agosto?

ALEJANDRO FARIAS
SALTO (BS AS)

Load MSX

El programa que mencionas está compuesto por tres bloques. El orden de copia no interesa pero recuerda hacer una copia en el disco de cada uno de ellos con el nombre que figura en el encabezamiento de cada listado: "balance.ini", "balance.ing" y "balance.cal". Luego se ejecuta "balance.ini" y por el menú de opciones del mismo se accede a los otros dos bloques. Una vez ejecutado "Balance.ing" se vuelve automáticamente al "balance.ini", y si se quiere que suceda lo mismo con "balance.cal" hay que agregar en este programa la línea 375 RUN"balance.ini"

CONSORCIOS

Luego de copiar el listado del programa "Consortios" y tratar de ejecutarlo encontré que el mismo se detiene por un error en la línea 2630. ¿Me podrían indicar cuál es?

VERONICA GAMBINA
Jose Ingenieros
(Bs As)

Efectivamente se deslizó en el listado un error en la línea que mencionas. Debe ir GOTO 2650 en vez de GOTO #.

Aprovechamos para comentar que la línea 2740 debe ser 2740 GOTO 290

COMPUTACION

K64

PARA TODOS

IMAGEN
DIGITALIZADA

NUEVAS

MSX-2 y D-C128
ARGENTINAS



D-C 64/128

HACKERS

SORTEO
Y
CONCURSO

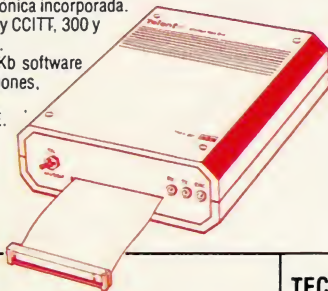
SPECTRUM-TK90
COMUNICACIONES

HARD y SOFT PARA ATARI, SPECTRUM, CZ, TK, DREAN COMMODORE, TI y MSX

Encienda una computadora **Talent** MSX y sus periféricos.

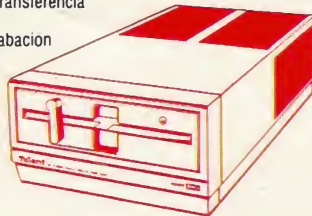
MODEM

- Interfaz asincrónica incorporada.
- Normas BELL y CCITT, 300 y 1.200 baudios.
- Incluye en 80 Kb software de comunicaciones, MSX-PLAN y MSX-WRITE.



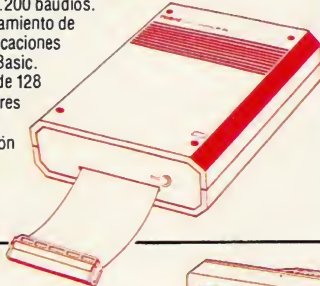
UNIDAD DE DISCOS FLEXIBLES

- 5 1/4" de 360 Kb (DS-DD)
- Velocidad transferencia 250 Kb/sg.
- Formato grabación compatible MS-DOS.



RS-232

- Velocidades programables desde 50 a 19.200 baudios.
- Procesamiento de comunicaciones desde Basic.
- Buffer de 128 caracteres para recepción



TECLADO NUMERICO



- Conexión a Joystick
- Se integra a todo el software que corre bajo MSX-DOS (Ej.: D-Base II, MSX-Plan, etc.)

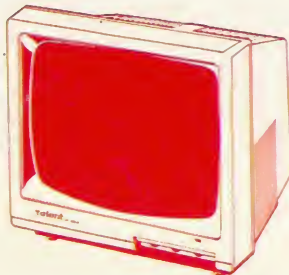
MOUSE

- Código DPM-220, accesorio para graficar.



MONITOR MONOCROMATICO 12"

- Anti-reflejo - Fósforo verde.
- Apto para uso profesional.
- (80 caracteres x 25 líneas).
- Parlante con amplificador incorporado.

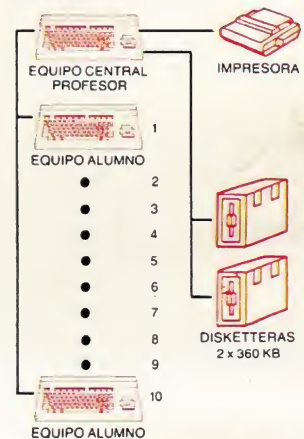


EXPANSION 80 COLUMNAS

- Hace posible la utilización de software estándar CP/M, emulando terminal tipo VT-52.
- Incluye software para manejo de video.



MINI-LAN



(RED PARA USO EDUCACIONAL)

- Comunicación por línea compartida a 30.000 baudios.
- Comparte unidades de disco e impresoras de consola MSX maestro con hasta 10 consolas MSX alumno
- Carga simultánea de un programa a todos los alumnos.
- Carga individual de maestro a alumno
- Salvado de programa alumno a unidad de disco maestro.
- Salida a impresora de maestro del listado de programa alumno, en spooling.
- Funciona desde MSX-Basic, MSX-Logo o cualquier programa que corra desde Basic.
- Estando activa, se dispone de todo el MSX-Basic.

Software

MSX-LOGO

Desarrollado por Logo Computer System Inc. con aplicación de primitivas y redacción del Manual por los Ings. Hilario Fernández Long y Horacio Reggini.

MSX-LPC

Lenguaje de programación estructurado y en castellano.

MSX-PLAN

Planilla de cálculo de Microsoft Corp. (Versión para MSX del Multiplan.)

MSX-WRITE

Procesador de palabra de ASCII Corp. en castellano.



Talent

Tecnología y Talento *en su casa*

Producida en San Luis por Telemática S.A. licenciataria exclusiva de Microsoft Corp. y ASCII Corp. para uso de la norma MSX en Argentina. 6 meses de garantía y mensualmente en su quiosco la revista Load MSX.

• MSX, MSX-DOS, MSX-PLAN, MS-DOS, son marcas registradas de Microsoft Corporation. MSX WRITE es marca registrada de ASCII Corporation.
• CP/M es marca registrada de Digital Research. MSX-LOGO es marca registrada de Logo Computer Systems Inc. Telemática: 1986. Todos los derechos reservados.
Los datos y especificaciones que figuran en este aviso pueden ser modificados sin previo aviso.